

Universidad Autónoma de Madrid

Escuela Politécnica Superior

Máster Universitario en Investigación e Innovación en TIC (i²-TIC)



Escuela Politécnica Superior

TRABAJO DE FIN DE MÁSTER

IMPLEMENTACIÓN DE UNA SOLUCIÓN INTERNET OF THINGS BASE PARA
FUTUROS DESARROLLOS DE APLICACIONES VERTICALES ENFOCADAS A
HACER EFICIENTES, OPTIMIZAR Y GESTIONAR ÁMBITOS O NEGOCIOS
CONCRETOS

Sonia Borondo Bellver

Directora: Rosa María Carro Salas

Madrid, Septiembre 2015

Agradecimientos

A Ramón, que se ha entregado a ayudarme en todo lo posible, siempre con buen humor y sin ninguna prisa, dedicándome todo el tiempo que he necesitado para resolver tantas y tantas dudas, sin él aún estaría perdida esbozando este trabajo.

A Javi, por estar siempre ahí, apoyándome en todo lo que he necesitado y ayudándome a potenciar cualquier idea que me proponga llevar a cabo, con la misma o mayor ilusión que la que tengo yo. Por su paciencia, su cariño y por sacar siempre lo mejor de mí.

A Miguel, por creer y confiar en mí e inculcarme que todo es posible si crees en ello, por las oportunidades que me ha dado y por servirme de ejemplo para demostrarme que no hay límites ni hay que rendirse con nada.

A Rosa, la directora de este trabajo que también fue mi profesora durante mis estudios de Ingeniería Informática, por ofrecerme siempre toda su ayuda y enseñarme cómo mejorar en cada tarea realizada. Ha sido muy gratificante trabajar juntas, me ha guiado durante todo el trabajo, corrigiéndome y enseñándome la forma de encontrar alternativas y soluciones a cada traba que me ha surgido.

Y especialmente, a mis padres y a mi hermano, les agradezco el poder contar siempre con ellos, en los momentos difíciles, son los que me ayudan y aconsejan para seguir adelante, además de involucrarme el amor por el estudio, las ganas de aprender y seguir evolucionando.

Resumen

Este trabajo se enmarca en el ámbito de la *Internet de las Cosas*, comúnmente conocido como *Internet of Things* (IoT). Los objetivos principales de este trabajo son dos: por una parte, idear y diseñar un prototipo de solución Internet of Things (IoT) integral, que abarque desde la captura de señales de los objetos físicos, pasando por las comunicaciones para el envío de datos entre los objetos y la nube, hasta el procesamiento e interpretación de la información enviada. Y por otro lado, llevar a cabo la implementación de la solución diseñada, con el objetivo de que sirva como prototipo en futuras soluciones verticales en cualquier sector o entorno concreto. Para ello, durante todo el trabajo, se ha intentado simplificar al máximo el diseño de la solución, de modo que resulte fácil de entender pero al mismo tiempo que abarque todas las etapas e incorpore todos los elementos necesarios para ofrecer una completa implementación de la misma.

Para satisfacer estos objetivos, se ha llevado a cabo una investigación sobre la tendencia tecnológica del IoT y del ecosistema M2M, compuesto por objetos físicos (móviles, sensores, chips RFID, tabletas, etc...) que están conectados y envían señales que son interpretadas, de modo que, mediante una inteligencia de negocio, se genera información que se envía al usuario final del sistema.

Por un lado, se ha llevado a cabo una labor de ingeniería para crear establecer el diseño de un prototipo de solución integral para IoT, ya que actualmente no existen metodologías ni arquitecturas predefinidas en las cuales basarse para la construcción de soluciones integrales. Por otro lado, se ha implementado la solución diseñada en la primera parte del trabajo. Esta solución se podrá utilizar como prototipo de partida para futuras soluciones verticales, ya que ha sido creada para aportar beneficios como el ahorro de tiempo en el análisis, diseño y desarrollo de este tipo de soluciones. Para crear una solución más compleja, basta con reutilizarlo y aumentar o escalar el diseño y la implementación como se desee.

Así como los objetivos iniciales de este trabajo se centraban en conocer en profundidad la tecnología M2M e IoT e investigar de qué manera se podría aportar algo novedoso y beneficioso en el mundo actual, la motivación personal final me ha conducido a crear una solución tangible de principio a fin, que sirva de base para construir otras soluciones IoT integrales, cubriendo así el hueco que existe actualmente en cuanto a metodologías, arquitecturas o prototipos concretos para este fin. Además, se describen en detalle dos ejemplos de aplicación concretos para dos contextos muy diferentes: la agricultura y producción inteligente del vino y un sistema de telemetría de boyas y estaciones meteorológicas.

Tabla de contenido

| | | |
|-------|--|----|
| 1 | Motivación y objetivos | 11 |
| 2 | Introducción a las soluciones IoT | 14 |
| 2.1 | Estructura de las soluciones IoT | 14 |
| 3 | Análisis de la cuestión | 16 |
| 3.1 | Enfoque metodológico | 16 |
| 3.2 | Módulos principales de una solución IoT | 17 |
| 3.2.1 | Las cosas | 17 |
| 3.2.2 | Procesamiento de la información/Interfaz de usuario | 18 |
| 3.2.3 | Módulo de conectividad..... | 20 |
| 3.3 | Conclusiones..... | 23 |
| 4 | Propuesta de diseño para la implementación de una solución IoT integral..... | 25 |
| 5 | Implementación de la solución | 27 |
| 5.1 | Dispositivos físicos..... | 27 |
| 5.1.1 | Nodo de sensores..... | 27 |
| 5.2 | Envío y recepción de datos | 30 |
| 5.2.1 | Interfaz WEB..... | 30 |
| 5.2.2 | API REST..... | 32 |
| 5.2.3 | Callbacks..... | 33 |
| 5.3 | Plataforma de desarrollo de la solución IoT..... | 33 |
| 5.3.1 | Desarrollo de la aplicación IoT sobre la plataforma Thingworx..... | 33 |
| 6 | Resultados y Aplicaciones | 44 |
| 6.1 | Valores actualizados en Nodo1 captados por el nodo de sensores real..... | 44 |
| 6.2 | Ejemplos concretos de ampliación de la solución | 45 |
| 6.2.1 | Agricultura y producción del vino inteligente | 45 |
| 6.2.2 | Sistema de telemetría de boyas y estaciones meteorológicas | 55 |
| 7 | Conclusiones y trabajo futuro | 62 |
| 8 | Bibliografía | 63 |

Figuras

| | |
|---|----|
| Figura 1 – Escenario actual del proceso | 13 |
| Figura 2 – Esquema simplificado de una solución IoT..... | 15 |
| Figura 3 - Diseño de la solución integral | 26 |
| Figura 4 – Tarjeta de conectividad con red Sigfox | 27 |
| Figura 5- Características de la tarjeta de conectividad con red Sigfox | 28 |
| Figura 6 – Placa de expansiones de sensores y módulo de conectividad | 28 |
| Figura 7 – Características de la placa de expansiones de sensores y módulo de conectividad.. | 29 |
| Figura 8 – Trama enviada por el dispositivo al Backend de Sigfox..... | 30 |
| Figura 9 – Recuperación de datos a través de la interfaz web de Sigfox..... | 31 |
| Figura 10 - Recuperación de datos a través de la API REST de Sigfox..... | 32 |
| Figura 11 – Esquema de elementos creados..... | 36 |
| Figura 12 - Servicio ActualizaLuminosidad | 37 |
| Figura 13 - Servicio RegistrarDatos | 40 |
| Figura 14 - Servicio LeerUltimoMensaje | 41 |
| Figura 15 - Tabla de datos históricos..... | 43 |
| Figura 16 – Gráfica de datos históricos | 43 |
| Figura 17 – Resultados. Valores actuales del nodo de sensores en Nodo1 | 44 |
| Figura 18 – Proceso completo de la agricultura y producción del vino | 45 |
| Figura 19 – Agricultura y producción inteligente del vino. Gestión de la información..... | 47 |
| Figura 20 - Agricultura y producción inteligente del vino. Monitorización y supervisión. | 47 |
| Figura 21 - Agricultura y producción inteligente del vino. Alertas..... | 48 |
| Figura 22 - Agricultura y producción inteligente del vino. Trazabilidad del proceso..... | 48 |
| Figura 23 - Agricultura y producción inteligente del vino. Control preventivo..... | 49 |
| Figura 24 - Agricultura y producción inteligente del vino. Agricultura de precisión..... | 49 |
| Figura 25 - Agricultura y producción inteligente del vino. Control de parámetros | 50 |
| Figura 26 - Agricultura y producción inteligente del vino. Almacenamiento y distribución..... | 50 |
| Figura 27 - Agricultura y producción inteligente del vino. Optimización del control de calidad | 51 |
| Figura 28 - Agricultura y producción inteligente del vino. Arquitectura de la solución. | 51 |
| Figura 29 - Agricultura y producción inteligente del vino. Vista cultivos..... | 53 |
| Figura 30 - Agricultura y producción inteligente del vino. Vista sala de producción..... | 53 |
| Figura 31 - Agricultura y producción inteligente del vino. Vista sala de producción con proceso detenido | 54 |
| Figura 32 - Agricultura y producción inteligente del vino. Cuadro de mando de la bomba de la sala de producción. | 55 |
| Figura 33 - Proceso del sistema de telemetría de boyas y estaciones meteorológicas | 56 |
| Figura 34 - Telemetría de boyas y estaciones meteorológicas. Monitorización y supervisión... | 57 |
| Figura 35 - Telemetría de boyas y estaciones meteorológicas. Alertas. | 58 |
| Figura 36 - Telemetría de boyas y estaciones meteorológicas. Trazabilidad del proceso | 58 |
| Figura 37 - Telemetría de boyas y estaciones meteorológicas. Control preventivo..... | 59 |
| Figura 38 - Telemetría de boyas y estaciones meteorológicas. Arquitectura de la solución..... | 59 |
| Figura 39 – Telemetría de boyas y estaciones meteorológicas. Vista estaciones. | 60 |

Figura 40 - Telemetría de boyas y estaciones meteorológicas. Vista cuadro de mando estación.
..... 61

Figura 41 - Telemetría de boyas y estaciones meteorológicas. Vista boyas. 61

1 Motivación y objetivos

Estamos en un momento en el que Internet of Things (IoT) no solo está transformando los productos y soluciones que ofrece la tecnología, sino que está cambiando por completo nuestro estilo de vida y la competencia entre empresas y compañías. El objetivo principal de este gran cambio, es aportar valor a la sociedad y a las empresas a través de la interacción entre Internet, las cosas y los datos.

Los hogares inteligentes son conscientes de lo que ocurre dentro de ellos, optimizando el uso de recursos (ahorro de agua, luz, gas, etc.), incrementando la seguridad (ante robos, incendios, etc.) y el confort. La intención es conseguir una mayor comodidad disminuyendo los gastos energéticos y por tanto, económicos. Así mismo, la transformación de ciudades a ciudades inteligentes (Smart cities) está potenciando el desarrollo sostenible y la calidad de vida. Algunas de las aplicaciones IoT existentes, como las que se emplean en los aparcamientos, sistemas de alumbrado, control de tráfico, etc. están disminuyendo el consumo energético y la contaminación, a la vez que aumentan la gestión de recursos y la seguridad en las calles.

Por otro lado, según el artículo “How Smart, Connected Products Are Transforming Competition” [3], estamos ante la tercera ola de IT que está impulsando la competitividad de las empresas. Esto se debe a la aparición de un nuevo concepto de los productos, en los que se empiezan a incorporar componentes físicos (partes mecánicas y eléctricas), componentes inteligentes (sensores, microprocesadores, S.O embebido, etc.) y componentes de conectividad (puestos, antenas, protocolos de conexión, etc.) para crear los “Smart, connected products”. La aparición de estos nuevos productos implican la modificación de la infraestructura tecnológica existente, que pasa a ser una nueva pila de tecnología con múltiples capas, incluyendo nuevo hardware, software embebido, conectividad, una nube de productos en la que el software se ejecuta en servidores remotos, un conjunto de herramientas de seguridad, un gateway para las fuentes de información externas, y la integración con los sistemas empresariales, como un CRM o un ERP.

Este gran cambio en la infraestructura y los productos actuales, nos ofrecen posibilidades como el monitoreo, el control, la optimización y la autonomía de las cosas en prácticamente cualquier tipo de contexto o situación real.

Consecuentemente, es imprescindible que nos planteemos qué impacto tiene este cambio en la estructura de la industria y la consecuente necesidad de redefinir los límites de la misma. Y nos hagamos preguntas sobre cómo los “smart, connected products” afectan a la cadena de valor o al conjunto de actividades requeridas para competir, qué nuevos tipos de opciones estratégicas se deben exigir a las empresas para lograr ventaja competitiva en los productos inteligentes o cuáles son las implicaciones organizativas de adoptar estos nuevos tipos de productos y los desafíos que afectan al éxito de la implementación.

Además, es muy importante tener presentes los errores a evitar en la competencia entre compañías que ofrecen productos o soluciones IoT. Por ejemplo, conviene valorar el riesgo de añadir una funcionalidad a un producto que los clientes no estén dispuestos a pagar, ya que el hecho de que sea posible ofrecer nuevas capacidades no significa que sea una propuesta de

valor clara para el cliente, ya que se puede llegar al punto en el que los rendimientos decrezcan debido al coste y a la complejidad de uso del nuevo producto. El no poder anticiparse a las nuevas amenazas de la competencia es otro factor a tener en cuenta, ya que si aparecen nuevos competidores que ofrecen productos con capacidades inteligentes (tales como la conectividad y el software embebido) o modelos de negocio basados en el rendimiento o en los servicios, éstos pueden emerger rápidamente y remodelar la competencia y las fronteras de la industria. Esperar demasiado tiempo para empezar permite a los competidores y nuevos participantes hacerse un hueco. Se debe evitar la sobreestimación de las capacidades internas, ya que los productos exigirán nuevas tecnologías, habilidades y procesos a lo largo de la cadena de valor (por ejemplo, análisis de datos masivos, ingeniería de sistemas, y desarrollo de aplicaciones de software). Es fundamental, por tanto, realizar una evaluación realista de las capacidades del producto que se deben desarrollar.

En la actualidad no existen metodologías, arquitecturas o prototipos que sirvan de base para construir soluciones IoT, que tengan en cuenta todos los elementos necesarios para que la solución sea integral, abarcando desde la recogida de información de los sensores hasta el envío y procesamiento de dicha información.

Los objetivos de este trabajo son dos: por una parte, realizar la labor de ingeniería para crear el diseño de un prototipo de una solución IoT integral, que abarque desde la captura de señales de los objetos físicos, pasando por las comunicaciones para el envío de datos entre los objetos y la nube, hasta el procesamiento e interpretación de la información enviada. Y por otro lado, llevar a cabo la implementación de la solución diseñada, con el objetivo de que sirva como prototipo para futuras soluciones verticales en cualquier sector o entorno concreto. El diseño de la solución propuesta pretende ser completo, abarcando todas las etapas y elementos necesarios para este tipo de soluciones, y a la vez sencillo, para facilitar su comprensión por parte de futuros desarrolladores en este ámbito. La figura 1 representa el escenario actual de la solución a diseñar e implementar. El primer paso/etapa consiste en capturar las señales a través de los sensores integrados en los objetos físicos. A continuación se captan los datos enviados por los sensores para procesarlos, analizarlos e interpretarlos. Finalmente se visualiza la información y se crean las alertas correspondientes.

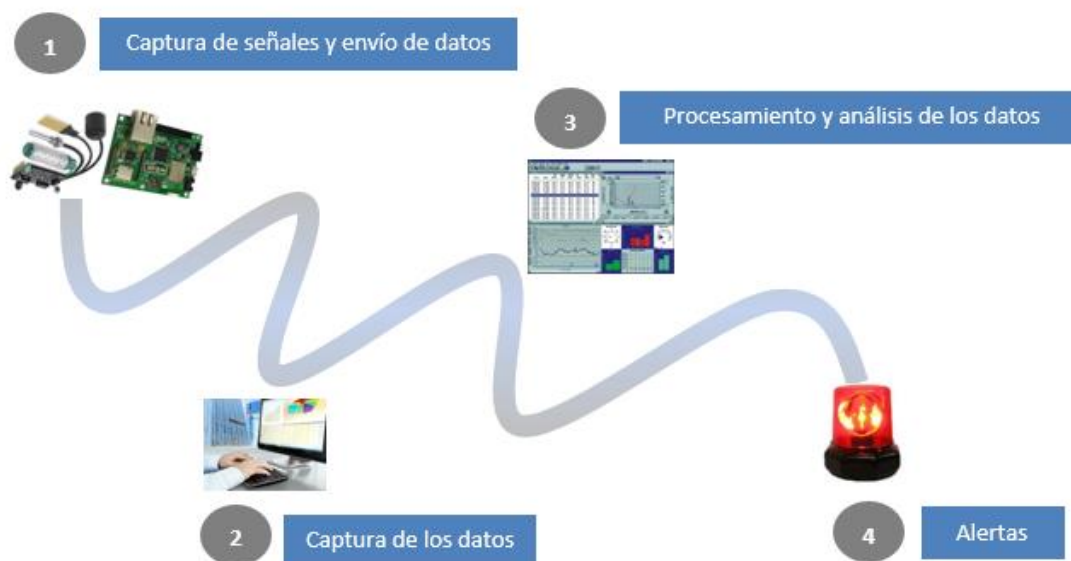


Figura 1 – Escenario actual del proceso

Este trabajo está estructurado de la siguiente manera. En la sección 2 se describen las características generales y los componentes que integran las soluciones verticales. En la sección 3 se presenta el enfoque metodológico seguido y se realiza un análisis de las características principales que debe poseer las “cosas” para pasar de ser objeto físicos a objetos inteligente, así como de otros atributos que pueden ser importantes en función del entorno donde se quiera implantar la solución. Después se establecen los criterios considerados para la elección de una plataforma de desarrollo de aplicaciones IoT y finalmente se realiza un estudio de los protocolos de red existentes para decidir cuál es el más adecuado utilizar en la solución a implementar.

En la sección 5, se detalla la propuesta y diseño de la solución creada. Tras el diseño, se explica la implementación de la solución en la sección 5.

La sección 6 recoge las pruebas y resultados obtenidos tras la implementación de la solución propuesta, e incluye la simulación de su aplicación en dos sectores muy concretos y diferentes entre sí.

Finalmente, en la sección 7 se presentan las conclusiones y las líneas de trabajo futuro. Por último, se incluye una sección con la bibliografía consultada.

2 Introducción a las soluciones IoT

La intervención humana en los procesos para el control, el seguimiento y el manejo de los objetos resulta una tarea laboriosa y costosa. Sin embargo, las soluciones de Internet of Things liberan a las personas de trabajos tediosos o peligrosos y reducen el coste al agilizar los procesos y realizarlos de forma más eficiente.

Las soluciones IoT son soluciones inteligentes que dotan a todo tipo de objetos físicos o dispositivos (móviles, sensores, chips RFID, tablets, etc...) de la capacidad de generar y recoger información, a través de unas señales que son interpretadas y que gracias a una inteligencia de negocio, generan eventos para enviar una respuesta al usuario final. Estas soluciones conectan datos con personas, procesos y objetos físicos a través de las actuales redes de comunicaciones (principalmente Internet). Aparecieron entre 2008 y 2009 en el momento en el que eran más las cosas conectadas a Internet que las personas y ha ido creciendo tanto su popularidad que actualmente estamos ante la tercera ola de IT en la que la unión de los componentes físicos, los componentes inteligentes y los componentes de conectividad han provocado la aparición de un nuevo concepto tecnológico, los “Smart, connected products”. A raíz de esta nueva era tecnológica han aparecido múltiples congresos y publicaciones relacionados con esta temática (IoT Solutions World Congress, IoT Week, IoT Forum, etc.).

2.1 Estructura de las soluciones IoT

En los procesos tradicionales, toda la información que manejaban las computadoras, dependía casi completamente de las personas, y éstas tienen limitaciones tanto en tiempo, exactitud y en la atención que esto involucra para capturar otro tipo de datos del mundo real. Además, se ha reflexionado sobre la necesidad de dotar de capacidad a las computadoras para ver, oír e incluso oler el mundo por ellas mismas ya que todo lo que nos rodea, incluidos nosotros mismos es físico [2].

Las soluciones IoT suplen la necesidad de intervención humana en la mayoría de los procesos, al convertir los objetos tradicionales en objetos inteligentes, conectados. Estos ecosistemas tecnológicos se basan en tres módulos:

- **Las cosas:** son el entorno físico. Van desde un objeto común, como unas gafas, hasta una máquina sofisticada, como la de montaje de piezas para automóviles. Incorporan dispositivos electrónicos como actuadores, microcontroladores y actuadores que dotan de “inteligencia” a las cosas. Son la fuente de información de la solución IoT a partir de la captura de señales del entorno en el que se encuentren.
- **Comunicaciones:** esta parte de la solución se encarga de la transmisión de señales recogida por los dispositivos electrónicos que tienen integradas las “cosas”. Aporta la capa de conectividad entre los objetos físicos y la nube a través de redes de comunicaciones.

- **Procesamiento/Interfaces de usuario:** es la capa de más alto nivel de una solución IoT. Procesa la información del entorno procedente de las cosas y la interpreta, de modo que, posteriormente, por un lado se envían datos a los objetos físicos que incorporan actuadores para que realicen acciones, y por otro se hace visible la información interpretada a través de diversas interfaces de usuario.

En la Figura 2 se muestra el esquema simplificado de una solución IoT, atendiendo a una definición muy amplia de la misma. Cada módulo es muy variopinto y no existe una solución común o predefinida, sino que depende de ámbito en el cual se quiera implantar. Se pueden crear soluciones IoT verticales que alcanzan prácticamente todos los sectores.

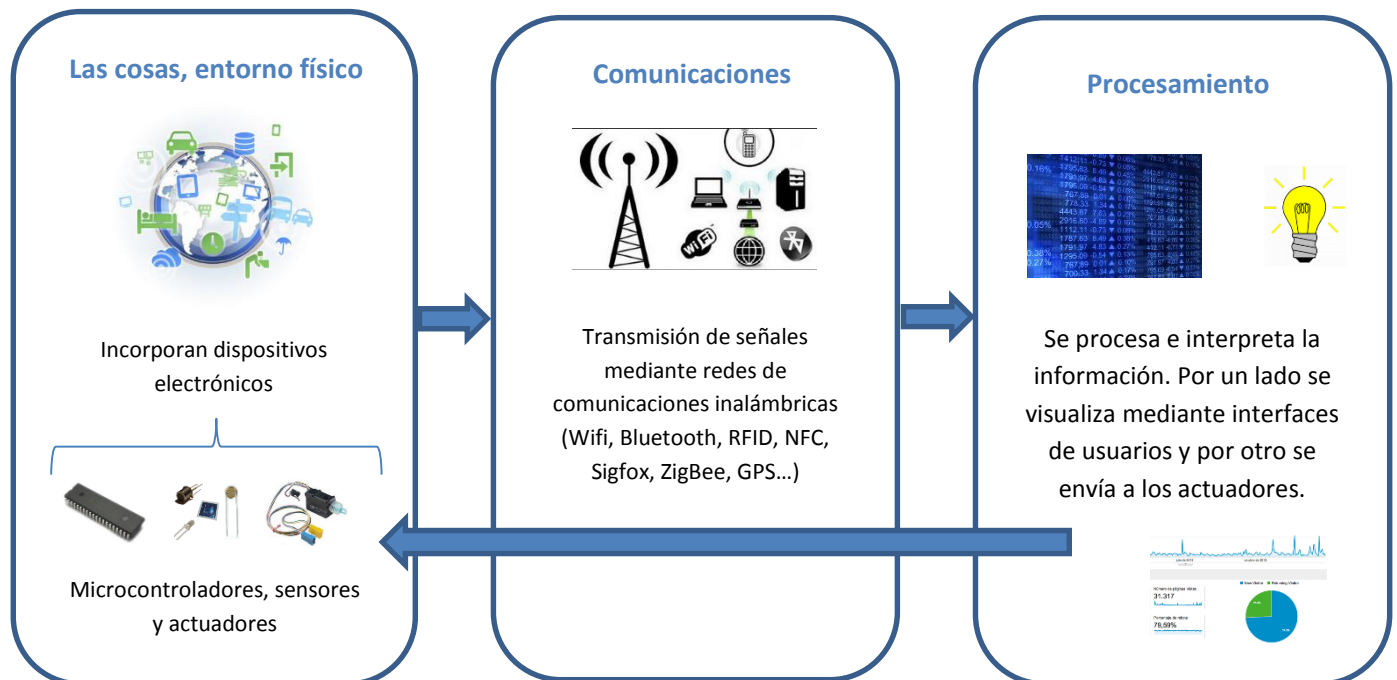


Figura 2 – Esquema simplificado de una solución IoT

Tras la búsqueda de metodologías y arquitecturas para la construcción de soluciones IoT desarrolladas hasta el momento, se puede concluir que no existe ninguna que establezca qué pautas se deben seguir o qué procedimientos se deben llevar a cabo para la construcción de una solución integral. Cada módulo de la estructura mostrada en la figura 2 tiene un gran abanico de posibilidades y la integración entre los mismos se puede realizar de múltiples maneras. Por lo que una de las aportaciones de este trabajo es la labor de ingeniería que se ha realizado para diseñar la solución propuesta sin contar con una metodología previamente establecida.

En la siguiente sección se describen los distintos elementos que pueden formar parte de estos módulos y se analizan las posibilidades existentes de cara a la elaboración de la solución propuesta.

3 Análisis de la cuestión

A partir de la estructura general de las soluciones IoT descrita en la sección anterior, se ha creado una solución IoT propia e integral, con el fin de que pueda ser utilizada como prototipo para soluciones verticales futuras y sectores o entornos concretos. De este modo, cuando se desee implantar una nueva solución, bastará con reutilizar el prototipo creado y ampliarlo o escalarlo tanto como se desee, puesto que en este trabajo ya se ha realizado la labor de ingeniería, diseño e implementación base.

El enfoque metodológico propuesto para abordar esta tarea está orientado, por tanto, al aporte o significación práctica, para ayudar y beneficiar a futuras implementaciones que se lleven a cabo. A continuación, en primer lugar se describe la propuesta de pasos a seguir para el desarrollo de la solución. Después se realiza un análisis detallado de los elementos que pueden formar parte de una solución IoT y se especifican las decisiones tomadas con respecto a cada uno de ellos, que influirán a su vez en el diseño e implementación de la solución propuesta.

3.1 Enfoque metodológico

Como se ha indicado anteriormente, ante la inexistencia de metodologías o arquitecturas previamente establecidas y reconocidas para el desarrollo de soluciones IoT, se propone un enfoque propio para abordar la creación de una solución con las características descritas en secciones anteriores. Este enfoque consta de las siguientes etapas:

- **Establecer los objetivos para la solución implementada:** Es fundamental conocer cuáles son las ventajas y beneficios que se quieren obtener con la solución implementada, para establecer los objetivos iniciales, y conocer las posibilidades de cada una de las partes de las soluciones IoT, para así definir los criterios más apropiados desde el comienzo del desarrollo.
- **Análisis y elección del módulo inicial de comienzo de la solución:** Se conoce la estructura general de las soluciones IoT (vista en la Figura 2), que está formada por tres módulos: las “cosas”, las comunicaciones y el procesamiento de la información, pero no existen pautas o procedimientos que guíen la construcción de la misma. Por ello, es necesario decidir el punto de partida en función de los objetivos establecidos para la solución implementada.
- **Integración y adecuación del resto de módulos:** Una vez elegido el módulo del que parte la implementación de la solución, se debe realizar un análisis de las características que deben tener los demás módulos a implementar para lograr una buena adaptación e integración entre los mismos.
- **Diseño y propuesta de la solución:** Crear el diseño y la propuesta de la solución en función de los resultados de los puntos anteriores.
- **Implementación de la solución:** Implementar la solución en base al diseño creado.

- **Pruebas de la solución:** Comprobar que la solución integral creada cumple los requisitos iniciales y que se obtienen los resultados esperados.
- **Ampliación de la solución:** Corroborar que el prototipo creado permite su reutilización en futuras soluciones verticales y mostrar qué beneficios les aporta la solución creada.

3.2 Módulos principales de una solución IoT

A continuación se describen los módulos principales de una solución IoT y se analizan las posibilidades existentes en cada uno de ellos, de cara a la elaboración del diseño de la solución propuesta.

3.2.1 Las cosas

Se refiere a la capa de más bajo nivel que conforma una solución IoT, las “cosas” del entorno físico que se quieren controlar y gestionar con un determinado fin.

Estas cosas pueden ser cualquier objeto físico que exista en nuestro mundo real: electrónica del hogar (tv, frigoríficos, lavadoras, alarmas de seguridad, detectores de paso...), dispositivos para hospitales (asistencia robótica para caminar, controlar el ritmo cardiaco, oxígeno en la sangre, temperatura, etc.), dispositivos geo-localizados (smartphones, localización por puntos WiFi, etc.), automoción (flotas, motores, elementos de seguridad, etc.), maquinaria industrial, alumbrado público, plazas de aparcamiento, animales, gafas y relojes inteligentes, etc.

¿Y cómo es posible que estas cosas nos aporten información y beneficios extra a los habituales? Es en este punto cuando los objetos de siempre pasan a ser nuevos conceptos. Para transformar una simple farola en una farola que sea capaz de activar encendidos/apagados y regular la intensidad de luminosidad, así como detectar si hay algún otro objeto próximo a ella, bastaría con integrar sensores de luminosidad, presencia, temperatura o humedad.

Todos los tipos de aparatos que nos rodean en nuestro mundo real, pueden ser modificados para trabajar en un sistema IoT. Por tanto, estamos ante un nuevo concepto de los productos, en los que se empiezan a incorporar componentes físicos (partes mecánicas y eléctricas), componentes “inteligentes” (sensores, microprocesadores, S.O embebido, etc.) y componentes de conectividad (puestos, antenas, protocolos de conexión, etc.) para crear los “Smart, connected products”.

Para que estos objetos físicos estén interconectados, es necesario estudiar y analizar los requisitos del sector en concreto en el que se quiere implantar una solución vertical de IoT. Por ejemplo, no son las mismas necesidades las que tiene una bodega de vinos que las que podría tener un sistema de telemetría de boyas y estaciones meteorológicas.

Para ello, durante el análisis realizado en este trabajo se han establecido las características principales que deben incorporar los objetos actuales para ser transformados en “Smart, connected products”.

3.2.1.1 Características principales a considerar para transformar un objeto en un producto inteligente y conectado.

Las características más importantes que se deben considerar sobre los dispositivos físicos que van integrados en “las cosas” de una solución IoT son las siguientes:

- **Conexión de sensores:** debemos analizar el número y tipo de diferentes sensores que es capaz de acoger el dispositivo. Un mismo dispositivo puede tener integrados numerosos sensores diferentes para aportar una gran cantidad de información de distintos tipos. Dependiendo de los parámetros que sean necesarios obtener, pueden ser sensores de: temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, movimiento, pH, etc.
- **Alimentación:** es importante conocer si el dispositivo va a encontrarse aislado y requiere de la capacidad de funcionar por batería (en cuyo caso se debe considerar la duración de la misma), o por el contrario va a estar siempre enchufado a una fuente de alimentación.
- **Conectividad:** una vez que los dispositivos captan las señales físicas a través de los sensores, se debe conocer cómo y dónde el dispositivo envía estas señales [16]. Para ello, hay que tener en cuenta la comunicación utilizada, es decir, a través de qué protocolo de red, el dispositivo envía los datos a la nube (Wifi, 2/3/4G, GPRS, GSM/SMS, ZigBee, Sigfox, etc.).

Otros atributos a tener en cuenta, que no todos los dispositivos tienen, son:

- **Sistema antimanipulación:** existen objetos que no deben poder manipularse, por motivos de seguridad o eficiencia, y otros cuyo manejo no afecta de ninguna forma al funcionamiento o rendimiento de éste.
- **Coeficiente de aislado** de la carcasa del dispositivo: puede ser imprescindible que el dispositivo tenga una protección especial y sea éste un requisito imprescindible a la hora de elegir el dispositivo.

3.2.2 Procesamiento de la información/Interfaz de usuario

Mientras que el apartado anterior se centra en la capa física de la solución IoT (las “cosas”), este apartado se dedica a cómo procesar, interpretar y visualizar la información recogida por los dispositivos. La creación de una aplicación que sirva como interfaz de usuario hará posible monitorizar y gestionar de forma fácil e intuitiva los objetos o “las cosas” involucrados en la solución.

Tras la captura de numerosos datos por parte del dispositivo (por ejemplo, valores de presión, de temperatura, de humedad, etc.) y su envío a la nube, ¿qué se puede hacer con estos datos?, ¿cómo se pueden tratar para obtener una información valiosa?

En algunos casos, basta con visualizar la información tal y como se recibe para obtener una información útil, pero, por lo general, se requiere de un procesamiento e interpretación de los datos para conseguir mayores beneficios en cada caso particular. A través de la combinación de la información proveniente de los distintos dispositivos y la aplicación de reglas, eventos, suscripciones, servicios, alarmas, etc. se puede incrementar notablemente la inteligencia de un sistema. De este modo, el usuario puede recibir no solo datos directos de los sensores, sino también información sobre lo que representan esos datos. Por otra parte, el hecho de que la aplicación creada disponga de una interfaz interactiva conectada con “las cosas”, permite al usuario no solo observar y monitorizar el estado de cada objeto, sino también interactuar con ellos, enviando información a los actuadores o interviniendo personalmente en algún punto de un proceso crítico del cual recibió una alarma, por ejemplo..

Además de ofrecer cierta inteligencia que permita interpretar los datos provenientes de los sensores, también es importante el modo en que se visualice la información. La interfaz para el usuario debe ser amigable e intuitiva, y la flexibilidad y riqueza de los elementos gráficos a utilizar, es otro factor a tener en cuenta para escoger una plataforma de desarrollo de aplicaciones de IoT.

3.2.2.1 Características principales para el desarrollo de la aplicación de una solución IoT.

Actualmente, hay numerosos frameworks para desarrollar aplicaciones de IoT con numerosas posibilidades que difieren entre unos y otros. Los rasgos principales que se han de tener en cuenta para la elección de una plataforma u otra, son los siguientes:

- **Desarrollo fácil, rápido y reutilizable:** Debido a que el objetivo es construir soluciones verticales adaptadas a cada sector, es muy importante que cada solución se pueda desarrollar de una manera sencilla y rápida, y que podamos reutilizar parte del desarrollo entre soluciones IoT. Por ejemplo, se podría utilizar o adaptar el desarrollo creado para controlar la luminosidad de una calle con el de un parking, modificando únicamente los rangos de luminosidad que establezca cada contexto.
- **Capa ubicua de conectividad:** necesitamos que las comunicaciones sean escalables, seguras, integrables y fácilmente desplegables para conectar sensores, dispositivos y equipos a través de cualquier topología de red y un escenario de comunicación.
- **Programación dirigida por eventos:** debido a que el motor principal de una solución IoT se basa en eventos, lo más adecuado es realizar un desarrollo en el que tanto la estructura como la ejecución de los programas vayan determinados por los sucesos que ocurran en los dispositivos, tanto definidos por el usuario o por los que sucedan automáticamente.
- **Despliegue de aplicaciones flexible:** esto permite que, en función de la disponibilidad de recursos del usuario y de las necesidades de sistemas del usuario, se pueda implantar la solución en la nube, en la locación/servidores del cliente o embebido en los propios dispositivos.

- **Conectividad y gestión de dispositivos:** al igual que hay que analizar la conectividad de los dispositivos (como se ha explicado en el apartado 3.2.1.1), hay que considerar la conectividad de la plataforma para recibir los datos, así como la seguridad de la conectividad.
- **Gestión de la información:** permite a los creadores de contenido conectar fácilmente con muchos tipos diferentes de sistemas dispares (como los sistemas ERP, sistemas de calidad, gestión de configuración de software, historiadores de procesos, etc.). Para ello, debe poder comunicarse con los protocolos y formatos de datos estándar (XML, JSON, CSV, etc.) y disponer de un driver (como JDBC) para comunicarse con cualquier base de datos back-end de la misma LAN / WAN. Esto no es sólo para la captura de datos, sino también para la exportación de datos del sistema.
- **Capacidad para establecer jerarquías y niveles de accesos:** permite definir reglas de visibilidad para permitir realizar acciones solamente a un usuario o grupo de usuarios o para permitir que múltiples grupos de usuarios visualicen el mismo modelo. Las estructuras jerárquicas permiten al usuario asignar visibilidad a las entidades del modelo. La visibilidad es una forma simple de control de acceso. Si a un usuario del sistema no se le concede visibilidad, entonces no existe dentro del dominio de ese modelo y por tanto, dicho usuario no puede realizar ninguna acción sobre el modelo.
- **Módulos de informes / “Accounting”:** Nº de dispositivos conectados a la plataforma, nº de sensores, nº de lecturas/peticiones/seg enviadas, datos de almacenamiento en Mb ocupados por el cliente, tiempo de permanencia de los datos on-line (1 años, 2 años,...) Ofrecer plantillas estándar para ser capaz de comunicarse con una amplia variedad de plataformas de configuración de dispositivos para lograr el aprovisionamiento y uso de los datos recogidos.
- **Soporte:** Disponer de un ecosistema de soporte (Web, manuales, ejemplos, foros....).

3.2.3 Módulo de conectividad

En los apartados 3.2.1 y 3.2.2, se han explicado la capa física o de bajo nivel y la capa de aplicación o de alto nivel, respectivamente. Para que ambas capas estén conectadas, debe existir otro nivel intermedio que permita la comunicación entre las “cosas” y la aplicación IoT [4]. Para conseguir dicha conectividad, es necesario el análisis comparativo de los protocolos de red existentes, explicitando qué necesidades debe cubrir la Internet of Things con respecto al modelo actual de la Internet de las personas.

3.2.3.1 Conectividad en Internet VS Conectividad en Internet of Things

Hasta ahora, el modelo actual de conectividad para Internet (de las personas) se ha centrado en cubrir y mejorar la descarga y compartición de grandes volúmenes de datos generados y recibidos por dispositivos tales como servidores, PCs de uso personal, tabletas, móviles, etc. Para ello, existen tecnologías como el 3G/4G o el ADSL/Fibra.

En Internet of Things cambia el modelo de conectividad actual, ya el objetivo que se persigue es que los objetos que estén conectados puedan ser cualquier cosa (un contenedor, una farola, una válvula,...) y las necesidades a cubrir durante la transmisión de datos dejan de ser el envío de un gran volumen de información en poco tiempo, como sucede actualmente en la Internet (de las personas). En la IoT se pretende que la transferencia de pequeñas cantidades de datos, correspondientes a simples cambios de estado de los objetos, tenga un coste energético lo más bajo posible, puesto que los objetos generalmente están aislados y no pueden tener una alimentación fija a la corriente de forma permanente.

Por ello, los paradigmas de las redes inalámbricas son diferentes para Internet y para Internet of Things.

Mientras que para las redes de internet móvil se centran principalmente en:

- Incrementar el ancho de banda (2/3/4G)
- Ofrecer tráfico de datos ilimitado
- Conseguir celdas de cobertura más pequeñas

Los requisitos de la red para IoT son:

- Bajo consumo energético
- Transmisión de pequeñas cantidades de datos limitadas
- Maximizar el alcance de la red
- Reducir costes

Es por tanto que la estrategia en IoT sigue otra línea diferente a la actual en Internet. Para cubrir las necesidades del nuevo modelo de conectividad en IoT se ha realizado una comparativa de los diferentes protocolos de conectividad inalámbrica existentes como Wi-Fi (IEEE 802.11), Bluetooth, ZigBee, GSM/UMTS/HSDPA, Sigfox, RFID/NFC etc.

Si se atiende a los cuatro objetivos que persigue la red de IoT, para un dispositivo físico que solo necesita enviar datos de la señales que capta, quedan descartadas las redes móviles por su alto coste y consumo energético. Por otro lado, la tecnología NFC no sería la más adecuada tampoco por su corto alcance.

Si finalmente se comparan las tecnologías Wi-fi, Bluetooth, ZigBee y Sigfox, pensando en grandes volúmenes de dispositivos conectados, que envían mensajes cortos y pueden estar situados a grandes distancias, Sigfox es el canal de transporte que más se adecua por su bajo consumo energético y sus bajos costes. Está diseñado para el envío de pequeñas cantidades de datos (mensajes de 123 bytes) a una baja velocidad (140 mensajes al día) en largas distancias. Es la que menor consumo energético realiza, además de tener costes económicos son muy bajos.

A continuación se da una breve descripción del uso de las tecnologías de comunicación inalámbricas existentes, que difieren de los objetivos de la utilización del protocolo de red Sigfox:

Wifi: diseñada para la transmisión de contenidos multimedia a unas tasas de velocidad altas en redes de área local.

Bluetooth: diseñada para la transmisión de contenidos multimedia (voz, datos, imágenes) a tasas de velocidad altas en redes personales.

ZigBee: creada para tener un bajo consumo y bajo coste a través de la transmisión robusta, sobre redes malladas, de pequeñas cantidades de datos, normalmente medidas de sensores o comandos de control para actuadores, en entornos industriales.

NFC: enfocada para comunicaciones de corto alcance (unos 10 cm) y alta frecuencia (13,56 MHz) permite el intercambio de datos entre dos recubrimiento NFC próximos.

Redes móviles: su principal ventaja es la alta velocidad para transmitir todo tipo de datos, en especial vídeos.

En el siguiente punto se explica en detalle este novedoso protocolo de comunicaciones Sigfox, pensado y enfocado a la red de IoT.

3.2.3.2 Protocolo de red Sigfox

Para la capa de conectividad de un dispositivo, Sigfox nos ofrece conectividad celular, a través de las amplias infraestructuras del grupo Abertis¹.

SIGFOX es un protocolo de red creado para el Internet de las cosas enfocado a comunicaciones de baja velocidad, lo que permite un menor consumo de energía así como reducir los costes de los dispositivos conectados.

Para conseguir esta gran ventaja de bajo consumo, utiliza la UNB (Ultra Narrow Band) de radiofrecuencia en la que se define un canal donde el ancho de banda del mensaje no excede significativamente el ancho de banda coherente del canal. La utilización de la UNB es esencial para suministrar una red de alta capacidad, evolutiva, de muy bajo consumo energético, conservando una infraestructura celular centrada en las estrellas simple y fácil de implantar.

Las bandas ISM (Industrial, Scientific and Medical) de radiofrecuencia electromagnética, son bandas reservadas internacionalmente para uso no comercial. Para la utilización de estas bandas es necesario respetar las regulaciones que limitan los niveles de potencia transmitida, pero el acceso es libre y no hay que poseer ninguna licencia. Comunicaciones como Wifi o Bluetooth hacen uso de esta banda.

La red funciona en las bandas ISM disponibles mundialmente (bandas de frecuencias sin licencia) y coexisten sobre estas frecuencias con otras tecnologías radio, pero sin ningún riesgo de colisión o de problemas de capacidad. SIGFOX utiliza actualmente la banda europea ISM más popular sobre 868MHz (tal como la define el ETSI y CEPT), así como la 902 MHz de EEUU (tal como la define la FCC), en función de las reglamentaciones regionales específicas.

¹ Abertis Infraestructuras, SA, es una corporación española fundada en 2003 con sede en Barcelona, dedicada a la gestión de autopistas e infraestructuras de telecomunicaciones.

En cuanto a la seguridad en las comunicaciones con SIGFOX, está ampliamente cubierta, a través de la protección anti-replay, mensaje de aleatorización, secuenciación, etc. La característica más importante de la seguridad de transmisión, sin embargo, es que solo los proveedores de dispositivos comprendan los datos intercambiados entre el dispositivo y los sistemas informáticos. SIGFOX actúa solamente como un canal de transporte, llevando los datos hacia el cliente del sistema IT.

Otro beneficio del uso de una tecnología con banda estrecha es el diseño de la antena, que permite mayor flexibilidad, ya que permite minimizar su tamaño y simplificarlas y sobre todo, son fácilmente personalizables y tienen un bajo coste sobre los aparatos donde se instalen.

Puede enviar entre 0 y 140 mensajes por día y cada mensaje puede contener hasta 12 bytes de datos reales de carga útil. El protocolo ya transmite el ID del dispositivo, de modo que los 12 bytes de datos son reales y no existe ninguna limitación de cómo estructurar estos 12 bytes.

Cada vez es mayor el número de plataformas IoT/M2M que soportan dicho protocolo, además de que puede ser usado en todos los emisores y receptores existentes.

3.3 Conclusiones

Tras analizar las características y posibilidades para cada uno de los tres grandes módulos que forman parte de la estructura de las soluciones IoT, se han ido cumpliendo los primeros pasos del enfoque metodológico propio descrito en la sección 3.1:

- **Establecer los objetivos para la solución implementada:** Se ha considerado que las ventajas y beneficios principales que se deben obtener al implantar la solución son: el bajo consumo energético, reducir costes económicos y que sea fácilmente escalable o ampliable a cualquier solución vertical de sectores concretos.
- **Análisis y elección del módulo inicial de comienzo de la solución:** De los tres módulos estudiados (las “cosas”, las comunicaciones y el procesamiento de la información de la Figura 2), el que más repercute en los objetivos establecidos en el punto anterior, es el módulo de comunicaciones. El consumo energético y los costes asociados a éste, gana protagonismo en la implementación de la solución respecto a los módulos de “las cosas” y el procesamiento de la información/interfaz de usuario. Esto se debe a que una solución IoT está pensada para crecer en número de dispositivos, y el consumo de energía y los costes de la solución, van en proporción a dicho crecimiento. Por ello, el diseño de la solución comienza con la elección del protocolo de red a utilizar y por tanto, con el diseño del módulo de comunicaciones. Tras la comparativa realizada en el apartado 3.2.3.1, se establece el protocolo Sigfox por ser el que mejor cumple las condiciones de bajo consumo energético, bajo coste y capacidad para abarcar un gran número de dispositivos situados a grandes distancias.

- **Integración y adecuación del resto de módulos:** Tras establecer el protocolo de red Sigfox para la capa de conectividad de la solución, se han de seleccionar dispositivos físicos, cuya configuración soporte dicho protocolo. Por otro lado, la plataforma para el procesamiento de la información y diseño de la interfaz de usuario debe permitir el acceso a la información del *backend* de Sigfox a través de la API REST (Ver 5.2.2). Así, tanto el módulo de “las cosas” como el módulo de procesamiento/interfaz de usuario, se han diseñado para adaptarse al módulo de comunicaciones.

A continuación, en las siguientes secciones, se describen los siguientes pasos dados durante el desarrollo de la solución: su diseño, su implementación, las pruebas realizadas y dos ejemplos de reutilización en soluciones verticales.

4 Propuesta de diseño para la implementación de una solución IoT integral

Una vez realizado el análisis de las distintas posibilidades para cada uno de los módulos y tomado las decisiones correspondientes, se presenta la siguiente propuesta de diseño de la solución integral objetivo de este trabajo.

Como se ha mencionado anteriormente, los tres módulos principales de la solución propuesta son:

- **Las cosas:** para este módulo se propone el uso de un nodo de sensores que permita la configuración del protocolo de red Sigfox.
- **Comunicaciones:** como se ha mencionado en los apartados previos, se ha elegido el protocolo de comunicaciones Sigfox, ya que es el que con un menor consumo energético y un bajo coste, permite llevar a cabo la solución y permitir la posibilidad de ampliarla en futuros trabajos o soluciones verticales.
- **Procesamiento/Interfaces de usuario:** para desarrollar este módulo se ha propuesto la plataforma ThingWorx² para el desarrollo de la aplicación, debido a que cubre perfectamente las características principales estudiadas en el punto 3.2.2.1 y permite la recuperación de datos de fuentes externas (como el *Backend* de Sigfox) en formato JSON a través de peticiones HTTP.

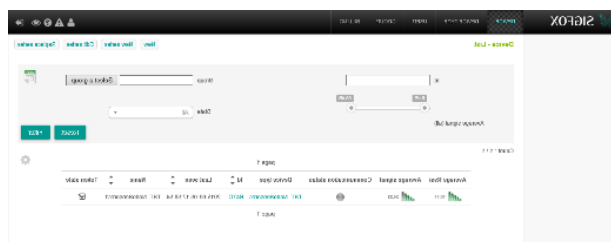
El esquema del diseño de la solución propuesta se muestra en la Figura 3: El nodo de sensores captura señales del entorno donde se sitúa y envía esta información en una trama, a través del protocolo Sigfox. En el backend de Sigfox se almacenan todas las tramas enviadas por los sensores, y permite que sean accesibles desde el exterior, a través de su API REST. En el otro extremo de la solución, se encuentra la aplicación desde la cual se construye la petición HTTP que se envía al backend de Sigfox y cuya respuesta se devuelve en formato JSON. Esta información es procesada e interpretada en un servicio creado en la aplicación, que desglosa la trama y actualiza los valores de cada propiedad recogida por los sensores (luminosidad, temperatura y humedad). Una vez obtenidos los datos desde la aplicación, se visualizan en la interfaz de usuario de manera que éste sea capaz de controlar y monitorizar la información recibida.

² <http://www.thingworx.com/>

Nodo de sensores con módulo de conectividad Sigfox



API REST del Backend de Sigfox



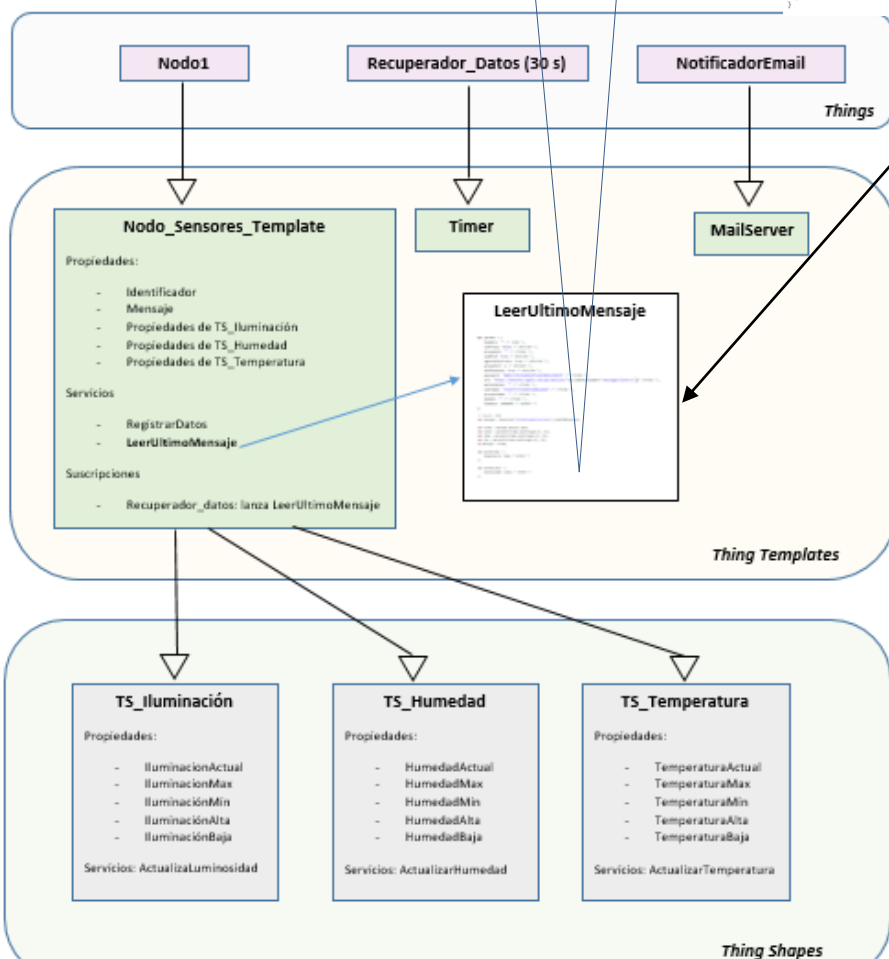
Petición HTTP

Respuesta JSON

`https://backend.sigfox.com/api/
/devices/" + me.Identificador + "
/messages?limit=1`

```
{
  "data": [ {
    "device": "788BCD",
    "time": 1442972104,
    "data": "0000110000",
    "seq": "24.58"
  } ],
  "paging": {
    "next": "https://backend.sigfox.com/api/devices/788BCD/messages?limit=1&before=1442972104"
  }
}
```

Modelo IoT



Se procesa el JSON y se actualizan los valores de los shapes TS_Iluminación, TS_Temperatura y TS_Humedad

Figura 3 - Diseño de la solución integral

Procesamiento de datos/Interfaz de usuario

5 Implementación de la solución

A partir del diseño de la solución propuesto en la sección anterior, donde se establecen los componentes del prototipo a construir y sus características, se ha desarrollado una implementación específica de la solución, que se describe a continuación.

Recordando una vez más la visión global de la solución, las tres partes fundamentales a integrar e interconectar son: dispositivos físicos (“las cosas”), aplicación con interfaz de usuario para el manejo de los dispositivos (“producto en la nube”) y capa de conectividad entre los dispositivos físicos y la aplicación de usuario.

5.1 Dispositivos físicos

Como se ha mostrado en el diseño de la solución, se ha seleccionado un nodo de sensores que incorpora un módulo de conectividad para soportar el protocolo Sigfox.

A continuación se muestran en detalle las especificaciones del dispositivo.

5.1.1 Nodo de sensores

El nodo de sensores está formado por los siguientes módulos:

- **Tarjeta de conectividad con red Sigfox** [6]: permite conectividad directa a la red Sigfox, tiene bajo consumo y es de fácil manejo y programación.



Figura 4 – Tarjeta de conectividad con red Sigfox

Las características específicas de la tarjeta son:

| ELÉCTRICO | |
|--------------|-------------------------|
| Alimentación | 2.3 - 3.3 VDC |
| Consumo | 13 - 16 mA RX, 37 mA TX |

| MECÁNICO | |
|-------------|---|
| Dimensiones | 51 mm alto / 145 mm diámetro |
| Conectores | Conector estándar de 2 filas compatible con sockets XBee™ |

| PARÁMETROS RADIO | |
|----------------------|------------------------|
| Bandas de frecuencia | 868 MHz |
| Sensibilidad | -126 dBm |
| Potencia de salida | 14 dBm |
| Certificaciones | Sigfox Class 0A device |

| SOFTWARE EMBEBIDO | |
|----------------------|---------------------------|
| Servicios soportados | Backend Sigfox |
| Comunicación | Comandos AT |
| | Librerías software de TST |

Figura 5- Características de la tarjeta de conectividad con red Sigfox

- **Placa base para las expansiones de sensores y el módulo de conectividad [7]:** A través de sus E/S y sus puertos serie se pueden conectar sensores, actuadores u otros dispositivos. Los sensores de luminosidad, temperatura y humedad, así como el módulo de conectividad Sigfox que se menciona en el punto anterior, van conectados en esta placa. Además, permite conectar otros módulos de expansión existentes con diversas tecnologías de comunicaciones. A nivel software incluye los protocolos TCP/IP, HTTP y Modbus, así como los drivers de los módulos de expansión. Corre sobre el sistema operativo en tiempo real FreeRTOS con soporte multitarea. Gracias a su diseño modular es posible combinar las tecnologías que se necesitan simplemente escogiendo el módulo de expansión correspondiente. La programación de dichos módulos es extremadamente sencilla con librerías software que dispone.

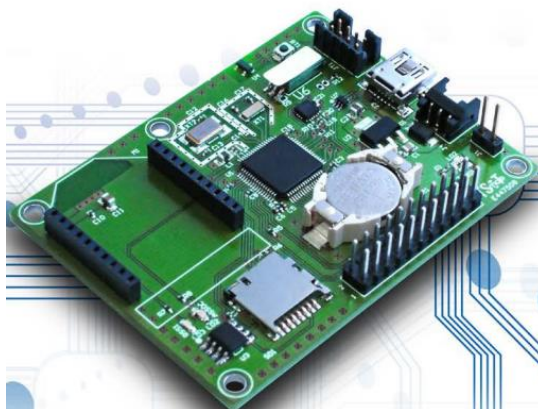


Figura 6 – Placa de expansiones de sensores y módulo de conectividad

Las características específicas del módulo son:

| ELÉCTRICO | |
|--|--|
| Voltaje de entrada | 4,5-12 VDC |
| Voltaje interno | 3,3 VDC |
| Consumo MCU On | 40 mA |
| Consumo MCU stand-by | 23 uA |
| Pila de botón | CR1225 |
| MECÁNICO | |
| Dimensiones | 70 x 52 mm |
| Conectores | Zócalo de 22 pines para módulos de expansión |
| | Zócalo de doble hilera para módulos de expansión |
| | Conector JTAG de 8 pines |
| | Mini USB |
| COMPUTACIÓN | |
| Microcontrolador | STM de 32 bits con núcleo ARM Cortex-M3 |
| Reloj | 72 MHz |
| Memoria Flash MCU | 1 MB |
| Memoria RAM MCU | 96 KB |
| Tarjeta SD | Slot para microSD de hasta 2 GB |
| Interfaces serie | 3 UART, 2 I2C, 1 SPI |
| Entradas / Salidas | Hasta 6 analógicas, hasta 20 digitales |
| MÓDULOS EXPANSIÓN | |
| ZigBee, Wi-Fi, IEEE802.15.4, GPRS, NFC/RFID, GPS, RS485, Sensores industriales | |
| SOFTWARE EMBEBIDO | |
| Protocolos TCP/IP, HTTP, Modbus | |
| Sistema operativo FreeRTOS | |
| Bibliotecas software de TST | |
| MEDIOAMBIENTAL | |
| Temperatura de funcionamiento | -20°C / +70°C |
| Temperatura de almacenaje | -40°C / +85°C |
| Certificaciones | CE, RoHS |

Figura 7 – Características de la placa de expansiones de sensores y módulo de conectividad

- **Caja IP65:** El nodo de sensores está cubierto por una caja con coeficiente de asilado IP65, que indica una protección fuerte contra el polvo (el polvo no debe entrar bajo ninguna circunstancia) y tiene protección contra chorros de agua (No debe entrar el agua arrojada a chorro (desde cualquier ángulo) por medio de una boquilla de 6,3 mm de diámetro, a un promedio de 12,5 litros por minuto y a una presión de 30 kN/m² durante un tiempo que no sea menor a 3 minutos y a una distancia no menor de 3 metros.)
- **Configuración de alimentación mixta** (220 VAC y batería recargable).

5.2 Envío y recepción de datos

La trama enviada por el nodo de sensores a través de Sigfox, en este caso solo ocupa 5 bytes de los 12 bytes disponibles, los datos vienen en hexadecimal y su estructura es la siguiente:

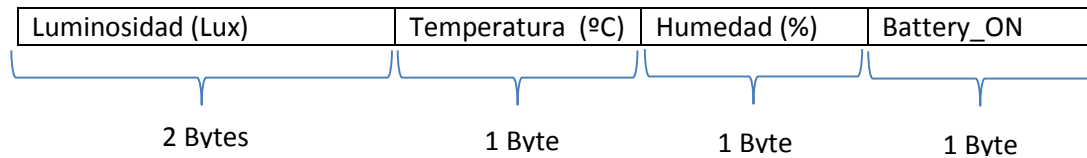


Figura 8 – Trama enviada por el dispositivo al Backend de Sigfox

- Luminosidad: Utiliza dos bytes y contiene el valor de la luminosidad capturado por dicho sensor y sus unidades son los Lux.
- Temperatura: Utiliza un byte y representa el valor de la temperatura capturada por el sensor. Se mide en °C.
- Humedad: Ocupa 1 byte que contiene el valor porcentual de la humedad medida por el sensor.
- Battery_ON: Se utiliza 1 byte para indicar si el dispositivo está consumiendo energía de la batería. Si su valor es 1, indica que sí está consumiendo energía de la batería; si por el contrario, su valor es 0, indica que no está haciendo uso de la batería y por tanto el dispositivo está conectado a una toma de red.

Una vez que el dispositivo envía el mensaje al *backend* de Sigfox, hay tres formas para obtener y utilizar los datos [8]:

- Desde la Interfaz Web en <http://backend.sigfox.com>
- A través de su API REST
- Mediante Callbacks

5.2.1 Interfaz WEB

Inicialmente, para comprobar que el nodo de sensores está captando las señales de luminosidad, temperatura, humedad y fuente de alimentación, y las está enviando correctamente, se accede al *backend* de Sigfox (<https://backend.sigfox.com>) para visualizar si están llegando los mensajes.

Para la realización de dicha prueba, se ha tenido encendido el dispositivo durante un largo periodo de tiempo, para comprobar por un lado, que se captan y envían señales por parte del dispositivo, y por otro, que dichos valores van cambiando con el paso del tiempo (ya que las condiciones de clima van cambiando según las diferentes horas del día).

A continuación se puede visualizar el resultado de dicha prueba y observar que efectivamente están llegando datos al *backend* y las tramas van cambiando con la evolución del tiempo.

Device B47C - Messages

| Time | Data / Decoding | Location | Signal (dB) | Callbacks |
|---------------------|----------------------------|----------|-------------|-----------|
| 2015-03-05 17:58:54 | 000b1a1800 ASCII: | | 24.46 | |
| 2015-03-05 17:48:54 | 000b1a1900 ASCII: | | 24.30 | |
| 2015-03-05 17:38:54 | 000b1a1900 ASCII: | | 23.21 | |
| 2015-03-05 17:28:54 | 000b1a1900 ASCII: | | 24.72 | |
| 2015-03-05 17:18:53 | 000b1a1900 ASCII: | | 24.12 | |
| 2015-03-05 17:08:53 | 000b1a1900 ASCII: | | 24.03 | |
| 2015-03-05 16:58:53 | 000b1a1a00 ASCII: | | 24.57 | |
| 2015-03-05 16:48:53 | 000b1a1a00 ASCII: | | 24.03 | |
| 2015-03-05 16:38:53 | 000b1a1a00 ASCII: | | 24.46 | |
| 2015-03-05 16:28:53 | 000b1a1a00 ASCII: | | 23.58 | |
| 2015-03-05 16:18:53 | 000b1a1a00 ASCII: | | 25.04 | |
| 2015-03-05 16:08:52 | 000b1a1a00 ASCII: | | 24.81 | |
| 2015-03-05 15:58:52 | 000b1a1a00 ASCII: | | 24.73 | |
| 2015-03-05 15:48:53 | 000b191900 ASCII: | | 23.07 | |

Figura 9 – Recuperación de datos a través de la interfaz web de Sigfox

Además, para comprobar que los valores que los sensores están capturando coinciden con las condiciones climatológicas de la sala donde se encuentra situado el dispositivo, se han de convertir los datos de la trama en hexadecimal a sus valores decimales correspondientes.

Si por ejemplo se toma la trama “000b191900” que aparece en la Figura 9, que corresponde a la captura de señales de las 15:48h, la división de la misma nos devuelve los siguientes valores:

- 000b 16) -> 11 Lux que corresponde al valor de la luminosidad en la sala.
- 19 16) -> 25 °C que corresponde a la temperatura de la sala.
- 19 16) -> 25% que corresponde a la humedad de la sala.
- 00 16) -> ON, indica que el dispositivo no está enchufado a la fuente de alimentación y está haciendo uso de la batería.

Si se toma la trama “000b1a1800” de la Figura 9, que corresponde a los datos recogidos a las 17:58h, se extraen los siguientes valores:

- 000b 16) -> 11 Lux que corresponde al valor de la luminosidad en la sala.
- 1a 16) -> 26 °C que corresponde a la temperatura de la sala.
- 18 16) -> 24% que corresponde a la humedad de la sala.
- 00 16) -> ON, indica que el dispositivo no está enchufado a la fuente de alimentación y está haciendo uso de la batería.

Se puede observar que la temperatura de la sala ha aumentado y la humedad se ha reducido.

Tras las pruebas anteriores se ha garantizado el correcto funcionamiento del nodo de sensores, así como la conectividad a través del protocolo Sigfox.

5.2.2 API REST

Como ya se ha mencionado anteriormente, es posible la recuperación de datos del *backend* de Sigfox a través de la API REST.

La API permite:

- Recuperar la lista de dispositivos asociados a un tipo de dispositivo.
- Recuperar los mensajes de un determinado dispositivo.
- Obtener métricas sobre los mensajes de un dispositivo.

Al igual que en el apartado anterior, se quiere comprobar que se están enviando mensajes cada diez minutos con los datos que captan los sensores. Para ello, se lanza la petición <https://backend.sigfox.com/api/devices/1B0CD/messages?limit=5> para recuperar por ejemplo las últimas cinco tramas (atributo "data") recibidas en el *backend*.

```
{
  "data" : [ {
    "device" : "1B0CD",
    "time" : 1441865993,
    "data" : "00911a1e00",
    "snr" : "24.67"
  }, {
    "device" : "1B0CD",
    "time" : 1441865392,
    "data" : "00611b1e00",
    "snr" : "24.52"
  }, {
    "device" : "1B0CD",
    "time" : 1441864792,
    "data" : "005b1b1e00",
    "snr" : "24.30"
  }, {
    "device" : "1B0CD",
    "time" : 1441864192,
    "data" : "005b1b1e00",
    "snr" : "23.66"
  }, {
    "device" : "1B0CD",
    "time" : 1441863592,
    "data" : "00591a1f00",
    "snr" : "23.86"
  } ],
  "paging" : {
    "next" : "https://backend.sigfox.com/api/devices/1B0CD/messages?limit=5&before=1441863592"
  }
}
```

Figura 10 - Recuperación de datos a través de la API REST de Sigfox

Como se puede observar los datos devueltos son en formato JSON. Este modo de recuperar la información a través de la API REST, es el que se utiliza en el desarrollo de la aplicación, que se expone en los siguientes apartados.

5.2.3 Callbacks

Además de las dos opciones de recuperación de datos de Sigfox vistas en los apartados anteriores, también es posible crear suscripciones para recibir una devolución de llamada HTTP por cada mensaje recibido y procesado por el *backend* SIGFOX. En este trabajo, las suscripciones se realizan desde el lado de la aplicación IoT, como se explicará en el punto 5.3.1.2.3.1.3.

5.3 Plataforma de desarrollo de la solución IoT

Una vez garantizado el correcto funcionamiento del nodo de sensores y la conectividad a través del protocolo Sigfox, comienza el desarrollo de la aplicación IoT que permite el procesamiento de los datos y la creación de la interfaz de usuario. Como se ha planteado en el diseño de la solución propuesto (Ver apartado **¡Error! No se encuentra el origen de la referencia.**) se ha seleccionado la plataforma ThingWorx para el desarrollo de la aplicación. Thingworx cubre las características principales estudiadas en el punto 3.2.2.1 y permite la recuperación de datos formato JSON procedentes de fuentes externas, vía HTTP, lo que permite el acceso al *backend* de Sigfox del mismo modo que se hace en el punto 5.2.2.

5.3.1 Desarrollo de la aplicación IoT sobre la plataforma Thingworx

Establecida la plataforma Thingworx para el desarrollo de la aplicación IoT, se definen los siguientes términos que corresponden a los elementos que se manejan en el modelador del framework.

5.3.1.1 Definición de elementos que conforman la solución

En la plataforma ThingWorx [9], cada aplicación IoT desarrollada es la construcción de un ecosistema formado, entre otros elementos, por objetos concretos que heredan de otros objetos genéricos las propiedades, servicios, eventos y suscripciones definidos en los mismos.

Por ejemplo, en el contexto o ecosistema de un Parking, los objetos concretos serían las plazas de aparcamiento: “Parking B32 Naranja”, “Parking A01 Azul”, etc. y el concepto de plaza de parking sería el objeto genérico, que tiene propiedades como humedad, temperatura, luminosidad etc., que heredan cada una de las plazas de parking específicas.

Si traducimos estos elementos al lenguaje que utiliza ThingWorx se corresponderían:

- Objetos concretos -> “Things”
- Objetos genéricos -> “Thing Templates
- Propiedades de los objetos -> “Thing Shapes”

Si asemejamos esta estructura a la de la programación orientada a objetos, un Thing Shape sería una clase abstracta, un Thing Template sería una clase y un Thing sería una instancia de dicha clase.

5.3.1.1.1 Things

Representan los objetos o cosas concretas que tienen propiedades y lógica de negocio en un contexto concreto. Además de configurar los Things para representar los dispositivos físicos, también se pueden utilizar para otras tareas basándose en ThingTemplates. Cuando se crea una cosa, se puede basar en una plantilla personalizada o una de las plantillas genéricas. Las propiedades estándar que un Thing posee son: descripción, nombre, estado, StatusMessage, etiquetas, las propiedades que herede de ThingTemplate, el evento DataChange y un conjunto estándar de servicios.

5.3.1.1.2 Thing Shapes

Como se ha explicado en el punto anterior, los Things son las cosas/objetos esenciales de la plataforma. Estos Things se crean generalmente para modelar los objetos físicos (máquinas, instalaciones, etc.), sistemas y dispositivos. Las cosas pueden ser bastante complejas y tener muchos aspectos diferentes. Un ThingShape es una definición abstracta de una cosa o cosas concretas. El ThingShape define propiedades, servicios, eventos y suscripciones para las cosas que implementan la ThingShape. Típicamente, el ThingShape se implementa por un ThingTemplate que es entonces la base de una cosa real. Otra metáfora de un ThingShape es un molde. Un molde puede ser utilizado para formar o hacer cosas tangibles, pero el propio molde no puede sustituir la cosa tangible. En el ejemplo mencionado al principio, los ThingShapes serían la humedad, luminosidad o temperatura.

5.3.1.1.3 Thing Templates

Los Things se derivan de ThingTemplates. Los ThingTemplates sirven para definir las propiedades, servicios, eventos y suscripciones que los *things*, o instancias de Things Templates, contendrán. Un Thing Template puede aplicar ninguno o más ThingShapes para ayudar en la reutilización de los objetos en el entorno creado. En el ejemplo mencionado al principio, el ThingTemplate sería el concepto de plaza de parking (no una plaza de parking en específica).

5.3.1.1.4 Otros elementos

5.3.1.1.4.1 Almacenamiento y manejo de datos

5.3.1.1.4.1.1 Data Shapes

Un Data Shape es una estructura de datos formada por uno o más campos, similar al diseño de una tabla de BBDD en la que cada campo contiene un nombre y un tipo de datos.

5.3.1.1.4.1.2 Streams

Un Stream es una lista de datos generados por los Things, basado en el tipo de Data Shape que se defina. Puede ser considerado como una estructura de tabla con cuatro campos predefinidos y otros campos definidos por el usuario. Ofrece una vista de los datos en modo tabla y modo gráfico, pudiendo ordenar y filtrar los datos.

5.3.1.1.4.2 Visualización

Una vez creado el ecosistema con todos los elementos y la lógica de negocio, viene una parte muy importante de cara al usuario final, la capa de visualización. Para construir dicha capa, Thingworx dispone de elementos de visualización que permiten un desarrollo muy flexible para reflejar cualquier contexto o entorno.

5.3.1.1.4.2.1 Mashups

Son las páginas web creadas para representar y/o interactuar con el modelo o ecosistema implementado [10]. Desde ellas se pueden gestionar y monitorizar los dispositivos conectados. Ejemplos de creación de estos Mashups se observan en las pantallas creadas para las soluciones verticales de la agricultura y producción inteligente del vino y en el sistema de telemetría de boyas y estaciones meteorológicas descritas en los apartados 6.2.1 y 6.2.2 respectivamente.

5.3.1.2 Elementos creados

En la Figura 11, se representan los elementos creados, junto a sus propiedades, servicios y suscripciones, así como las relaciones de herencia entre los diferentes elementos.

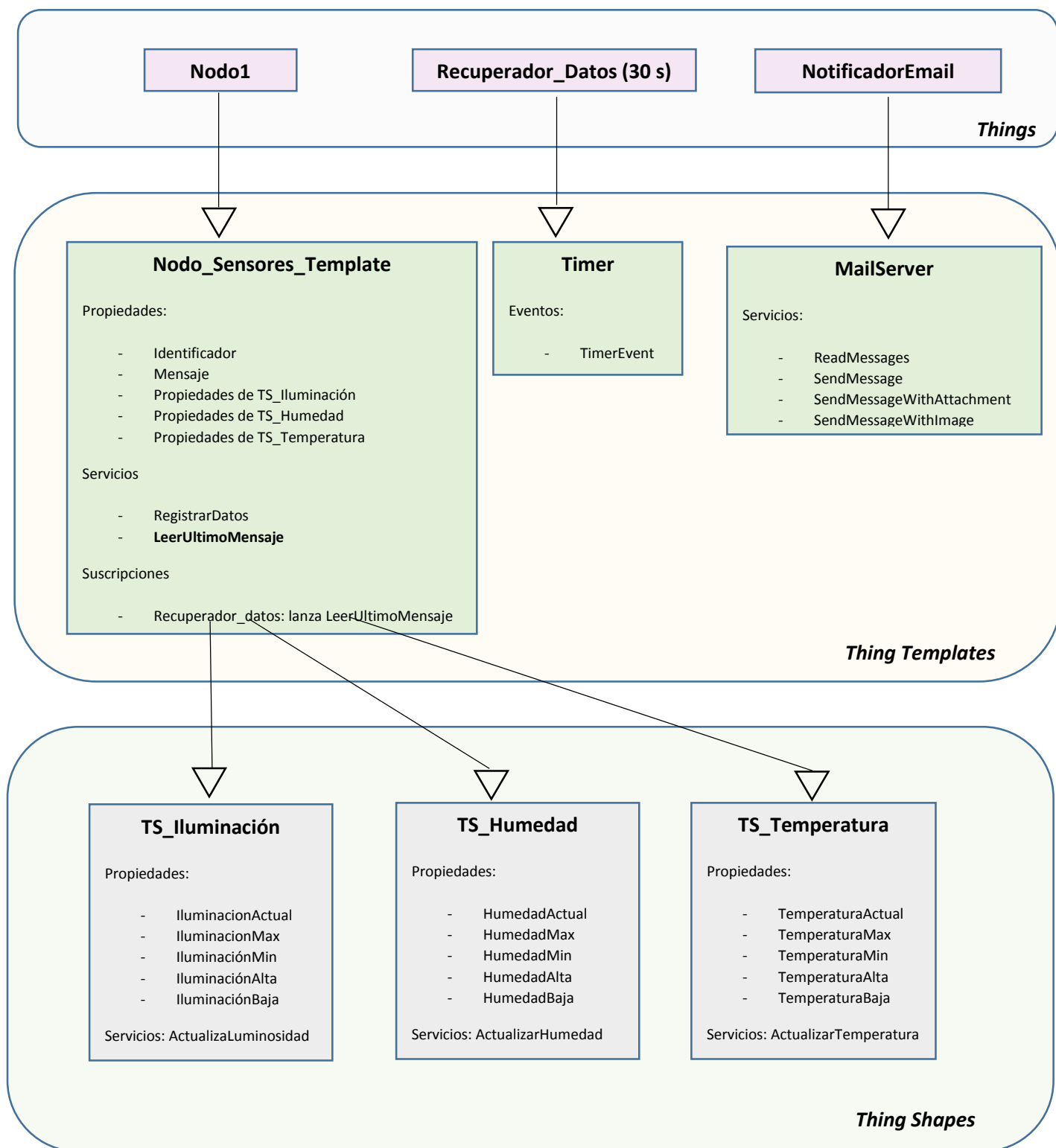


Figura 11 – Esquema de elementos creados.

5.3.1.2.1 Things shapes

Para cada medida que recoge el nodo de sensores: luminosidad, temperatura, humedad y uso de batería, se ha creado un Thing Shape y se han definido, para cada uno de ellos, las propiedades y servicios que se indican a continuación.

5.3.1.2.1.1 TS_Iluminación

Propiedades:

- IluminaciónActual (number): valor de la luminosidad actual en Lux recogida por el sensor.
- IluminaciónMax (number): valor de la luminosidad máxima permitida en Lux.
- IluminacionMin (number): valor de la luminosidad mínima permitida en Lux.
- IluminacionAlta (boolean): verdadero si IluminaciónActual > IluminacionMax, falso en caso contrario.
- IluminacionBaja (boolean): verdadero si IluminaciónActual < IluminacionMin, falso en caso contrario.

Servicios:

- ActualizaLuminosidad: Actualiza las propiedades anteriores con el valor del input “luminosidad”. El script del servicio es el siguiente:

```
//Se actualizan los valores de luminosidad

me.IluminacionActual = luminosidad;
if (luminosidad >= me.IluminacionMax){
    me.IluminacionAlta = true;
}
else{
    me.IluminacionAlta = false;
}

if (luminosidad <= me.IluminacionMin){
    me.IluminacionBaja = true;
}else{
    me.IluminacionBaja = false;
}
```

Figura 12 - Servicio ActualizaLuminosidad

5.3.1.2.1.2 TS_Temperatura

Propiedades:

- TemperaturaActual (number): valor de la temperatura actual en °C recogida por el sensor.
- TemperaturaMax (number): valor de la temperatura máxima permitida en °C.

- TemperaturaMin (number): valor de la temperatura mínima permitida en °C.
- TemperaturaAlta (boolean): verdadero si TemperaturaActual > TemperaturaMax, falso en caso contrario.
- TemperaturaBaja (boolean): verdadero si TemperaturaActual < TemperaturaMin, falso en caso contrario.

Servicios:

- ActualizarTemperatura: Actualiza las propiedades anteriores con el valor del input “temperatura”. El script del servicio es similar al del servicio ActualizaLuminosidad de la Figura 12, pero con los parámetros correspondientes a la temperatura y los límites que determinan cuando la temperatura es alta y cuándo baja.

5.3.1.2.1.3 TS_Humedad

Propiedades:

- HumedadActual (number): valor de la humedad actual en % recogida por el sensor.
- HumedadMax (number): valor de la humedad máxima permitida en %.
- HumedadMin (number): valor de la humedad mínima permitida en %.
- HumedadAlta (boolean): verdadero si HumedadActual > HumedadMax, falso en caso contrario.
- HumedadBaja (boolean): verdadero si HumedadActual < HumedadMin, falso en caso contrario.

Servicios:

- ActualizarHumedad: Actualiza las propiedades anteriores con el valor del input “humedad”. El script del servicio es similar al del servicio ActualizaLuminosidad de la Figura 12, pero con los parámetros correspondientes a la humedad y los límites que determinan cuando la humedad es alta y cuándo baja.

5.3.1.2.2 Things

5.3.1.2.2.1 Recuperador_Datos

Este Thing es un temporizador (extiende del Thing Template “Timer”) y ha sido configurado con un valor de 30 segundos. La utilidad del mismo es que el objeto genérico “Nodo_Sensores_Template” contenga un servicio para actualizar los datos enviados por el nodo de sensores en la aplicación y se suscriba a este Thing para que se llame a dicho servicio cada 30 segundos.

5.3.1.2.2.2 *NotificadorEmail*

Contiene la configuración del correo para el envío de notificaciones y alarmas.

5.3.1.2.2.3 *Nodo1*

Representa el nodo de sensores físico. Hereda del Thing Template “Nodo_Sensores_Template” (ver 5.3.1.2.3.1) por lo que posee todas las propiedades del mismo (mensaje, luminosidad, temperatura, humedad y uso de batería).

5.3.1.2.3 Thing Templates

5.3.1.2.3.1 *Nodo_Sensores_Template*

Este es el objeto genérico que representa un nodo de sensores y cuyas propiedades, servicios y suscripciones heredará cualquier nodo de sensores concreto que se conecte al sistema.

Dicho nodo se crea en este modo plantilla u objeto genérico ya que, aunque en esta solución se dispone de un único dispositivo, se debe implementar de cara a que en el futuro se pueda ampliar el número de dispositivos conectados y dicho objeto pueda ser reutilizado tantas veces como nuevos dispositivos se conecten a la solución.

5.3.1.2.3.1.1 *Propiedades*

Dado que el dispositivo físico captura las señales de luminosidad, temperatura, humedad y uso de batería, se han asociado los Thing Shapes TS_Iluminacion, TS_Humedad, TS_Temperatura y TS_BateriaON a este objeto.

De este modo, las propiedades que posee Nodo_Sensores_Template son todas las propiedades de los Thing Shapes asociados (por ejemplo, de TS_Iluminacion, heredará las propiedades de IluminaciónActual, IluminaciónMin, IluminaciónMax, etc.), además de las siguientes que se han definido:

- Identificador (String): contiene el identificador del nodo de sensores físico.
- Mensaje: (String), guarda la trama extraída de Sigfox.

5.3.1.2.3.1.2 Servicios

- RegistrarDatos: Vuelca al histórico de datos los valores actuales recogidos por el sensor. Éste es llamado desde el servicio “LeerUltimoMensaje, tras capturar los datos del *backend*. El script del servicio se muestra en la Figura 13.

```
var values = Things["StreamNodoSensores"].CreateValues();

//values.Mensaje = "aaa"; //STRING
values.Luminosidad = me.IlluminacionActual; //NUMBER
values.Humedad = me.HumedadActual //NUMBER
values.Temperatura = me.TemperaturaActual //NUMBER

var params = {
  tags: undefined /* TAGS */,
  timestamp: new Date() /* DATETIME */,
  sourceType: "Nodo" /* STRING */,
  values: values /* INFOTABLE*/,
  source: me.name /* STRING */,
  location: me.location /* LOCATION */
};

Things["StreamNodoSensores"].AddStreamEntry(params);
```

Figura 13 - Servicio RegistrarDatos

- LeerUltimoMensaje: Recupera del *backend* de Sigfox via HTTP mediante la API REST, los valores en formato JSON. Además, se procesa la información y se actualizan las propiedades anteriores (valores de la trama, luminosidad, temperatura, humedad y uso de la batería) de todos los things que ehereden de Nodo_Sensores_Template (en este caso, solo existe el thing “Nodo1”). El script del servicio se muestra en la Figura 14:


```

if(me.Identificador.length != 0){
    var params = {
        headers: "" /* JSON */,
        useProxy: false /* BOOLEAN */,
        proxyHost: "" /* STRING */,
        useNTLM: true /* BOOLEAN */,
        ignoreSSLErrors: true /* BOOLEAN */,
        proxyPort: 1 /* INTEGER */,
        withCookies: true /* BOOLEAN */,
        password: "8885c5703fed8dc0f13dd3005e7686f6" /* STRING */,
        url: "https://backend.sigfox.com/api/devices/"+me.Identificador+"/messages?limit=1"
/* STRING */,
        workstation: "" /* STRING */,
        username: "552d47fc933682fa90e2282d" /* STRING */,
        proxyScheme: "" /* STRING */,
        domain: "" /* STRING */,
        timeout: 1000000 /* NUMBER */
    };
    // result: JSON
    var mensaje = Resources["ContentLoaderFunctions"].LoadJSON(params);

    var trama = mensaje.data[0].data;
    var lumin = parseInt(trama.substring(0,4), 16);
    var temp = parseInt(trama.substring(4,6), 16);
    var hum = parseInt(trama.substring(6,8), 16);
    me.Mensaje = trama;

    var paramsTemp = {
        temperatura: temp /* NUMBER */
    };

    var paramsLumin = {
        luminosidad: lumin /* NUMBER */
    };

    var paramsHum = {
        humedad: hum /* NUMBER */
    };

    me.ActualizarTemperatura(paramsTemp);
    me.ActualizaLuminosidad(paramsLumin);
    me.ActualizarHumedad(paramsHum);

    if(parseInt(trama.substring(8,10), 16) == 1){
        me.BateriaOn = false;
    }else{
        me.BateriaOn = true;
    }
    logger.warn(trama);
    logger.warn("Luminosidad: "+trama.substring(0,4)+" - "+lumin);
    logger.warn("Humedad: "+trama.substring(6,8)+" - "+hum);
    logger.warn("Temperatura: "+trama.substring(4,6)+" - "+temp);
    logger.warn("Bateria: "+trama.substring(8,10)+" - "+me.BateriaOn);
    logger.warn("Recuperación terminada");
    me.RegistrarDatos();
}

```

Figura 14 - Servicio LeerUltimoMensaje

5.3.1.2.3.1.3 *Suscripciones*

Las suscripciones creadas para que se ejecute el script con la petición a la API REST de Sigfox de forma periódica son las siguientes:

- Recuperador_datos (Ver 5.3.1.2.2.1): Suscripción al thing “Recuperador_datos” para que cada 30 segundos, se ejecute el servicio LeerUltimoMensaje.

5.3.1.2.4 *Data Shapes*

5.3.1.2.4.1 *DS_NodoSensores*

Estructura con los tipos de datos capturados por los sensores. Los campos definidos son:

- Mensaje (String): Trama completa que envía el nodo de sensores.
- Luminosidad (Number): Valor de la luminosidad captada.
- Temperatura (Number): Valor de la temperatura captada.
- Humedad (Number): Valor de la humedad captada.

5.3.1.2.5 *Streams*

5.3.1.2.5.1 *StreamNodoSensores*

Tabla basada en el tipo DS_NodoSensores que almacena el histórico de los valores capturados por el nodo de sensores. En la Figura 15 se muestran los datos en modo de tabla. Por ejemplo, la primera línea representa que el día 11 de septiembre el objeto ‘Nodo1’ estaba situado en un entorno que tenía 27°C de temperatura, 31% de humedad y 48 Lux de luminosidad. Como se puede observar, durante los tres minutos que se muestran en la imagen, las condiciones de luminosidad, humedad y temperatura del Thing *Nodo1*, se mantienen sin cambios.

Search: Source: Tags

2014-09-11 05:54:41 2015-09-11 05:54:41

Update Data

None of 10 Active Filter(s)

Raw Data Chart

| Mensaje | Timestamp | Tags | Id | SourceType | Source | Location | Luminosidad | Humedad | Temperatura |
|------------|---------------------|------|--------|------------|--------|-------------|-------------|---------|-------------|
| 00301b1f00 | 2015-09-11 04:47:31 | | 403520 | Thing | Nodo1 | 0.00 : 0.00 | 48 | 31 | 27 |
| 00301b1f00 | 2015-09-11 04:47:01 | | 403516 | Thing | Nodo1 | 0.00 : 0.00 | 48 | 31 | 27 |
| 00301b1f00 | 2015-09-11 04:46:31 | | 403512 | Thing | Nodo1 | 0.00 : 0.00 | 48 | 31 | 27 |
| 00301b1f00 | 2015-09-11 04:46:01 | | 403508 | Thing | Nodo1 | 0.00 : 0.00 | 48 | 31 | 27 |
| 00301b1f00 | 2015-09-11 04:45:31 | | 403504 | Thing | Nodo1 | 0.00 : 0.00 | 48 | 31 | 27 |
| 00301b1f00 | 2015-09-11 04:45:01 | | 403499 | Thing | Nodo1 | 0.00 : 0.00 | 48 | 31 | 27 |
| 00301b1f00 | 2015-09-11 04:44:31 | | 403495 | Thing | Nodo1 | 0.00 : 0.00 | 48 | 31 | 27 |

Figura 15 - Tabla de datos históricos

La visualización de los datos en modo gráfico se muestra en la Figura 16. Las líneas de colores representan los valores recogidos por los sensores. La verde corresponde a la humedad, la roja a la temperatura y la azul a la luminosidad.



Figura 16 – Gráfica de datos históricos

6 Resultados y Aplicaciones

A continuación se describen los resultados obtenidos tras la implementación de la solución integral, en los que se muestran los valores recogidos por el nodo de sensores físico, a través del *thing* “Nodo1” de la aplicación. Más adelante, en la sección 6.2 se describen dos ejemplos concretos de ampliación de la solución IoT, aplicados a dos sectores concretos (agricultura y producción inteligente del vino y un sistema de telemetría de boyas y estaciones meteorológicas)

6.1 Valores actualizados en Nodo1 captados por el nodo de sensores real

Tras la creación de los elementos necesarios y el desarrollo de la lógica de negocio, en la plataforma de desarrollo se pueden observar los valores de los distintos parámetros de cada uno de los objetos que conforman el sistema. Por ejemplo, la Figura 17 muestra los valores reales de las señales de luminosidad, temperatura, humedad y uso de la batería que se están capturando para el Thing *Nodo1* en un instante determinado.

| Properties | | | | | | | | | |
|--|------|-----------------|---------------|------------|-------------|--|--|--|--|
| ▼ My Properties | | | | | | | | | |
| Name | Type | Additional Info | Default Value | Value | DataChange: | | | | |
| No Properties | | | | | | | | | |
| ▼ Nodos_Sensores_Template (ThingTemplate) - Properties | | | | | | | | | |
| Name | Type | Additional Info | Default Value | Value | DataChange: | | | | |
| Mensaje | - | | | 005b1b1d00 | VALUE | | | | |
| Identificador | - | | | 180CD | VALUE | | | | |
| ▼ TS_Temperatura | | | | | | | | | |
| Name | Type | Additional Info | Default Value | Value | DataChange: | | | | |
| TemperaturaActual | # | °C | | 27.0 | VALUE: 0 | | | | |
| TemperaturaMax | # | °C | 25.0 | 20.0 | VALUE: 0 | | | | |
| TemperaturaMin | # | °C | 15.0 | 15.0 | VALUE: 0 | | | | |
| TemperaturaAlta | | | false | true | VALUE | | | | |
| TemperaturaBaja | | | false | false | VALUE | | | | |
| ▼ TS_BateriaON | | | | | | | | | |
| Name | Type | Additional Info | Default Value | Value | DataChange: | | | | |
| BateriaOn | | | | true | VALUE | | | | |
| ▼ TS_Iluminacion | | | | | | | | | |
| Name | Type | Additional Info | Default Value | Value | DataChange: | | | | |
| IluminacionActual | # | lux | 500.0 | 91.0 | VALUE: 0 | | | | |
| IluminacionMax | # | lux | 1000.0 | 1000.0 | VALUE: 0 | | | | |
| IluminacionMin | # | lux | 200.0 | 200.0 | VALUE: 0 | | | | |
| IluminacionAlta | | | false | false | VALUE | | | | |
| IluminacionBaja | | | false | true | VALUE | | | | |
| IluminacionUmbral | # | | 500.0 | 500.0 | VALUE: 0 | | | | |
| IluminacionActiva | | | false | true | VALUE | | | | |
| ▼ TS_Humedad | | | | | | | | | |
| Name | Type | Additional Info | Default Value | Value | DataChange: | | | | |
| HumedadActual | # | % | 50.0 | 29.0 | VALUE: 0 | | | | |
| HumedadMax | # | % | 85.0 | 85.0 | VALUE: 0 | | | | |
| HumedadMin | # | % | 20.0 | 20.0 | VALUE: 0 | | | | |
| HumedadAlta | | | false | false | VALUE | | | | |
| HumedadBaja | | | false | false | VALUE | | | | |

Figura 17 – Resultados. Valores actuales del nodo de sensores en Nodo1

6.2 Ejemplos concretos de ampliación de la solución

El prototipo construido en este trabajo, en el que se observa cómo un dispositivo físico, en este caso, el nodo de sensores, captura señales, envía datos reales a la nube y estos se ven reflejados en un objeto virtual, constituye una base sólida y un diseño establecido para el comienzo del desarrollo de soluciones verticales en las que bastará con reutilizar el prototipo y ampliar o escalar la solución al número y tipo de dispositivos que se desee.

Para mostrar cómo se aplicaría la solución integral implementada en el presente trabajo, se han elaborado dos escenarios en dos contextos diferentes. El primer ejemplo se trata de una solución vertical para gestionar y monitorizar la agricultura y producción del vino de forma inteligente. La segunda aplicación presenta un sistema de telemetría de boyas y estaciones meteorológicas. En los siguientes apartados se explica en detalle cada una de las soluciones verticales.

6.2.1 Agricultura y producción del vino inteligente

6.2.1.1 Escenario actual

Tras realizar un estudio de negocio para aprender cómo funciona una bodega para la producción de vino, se obtiene que las etapas del proceso completo que influyen en la calidad del producto, mostradas en la Figura 18, son:

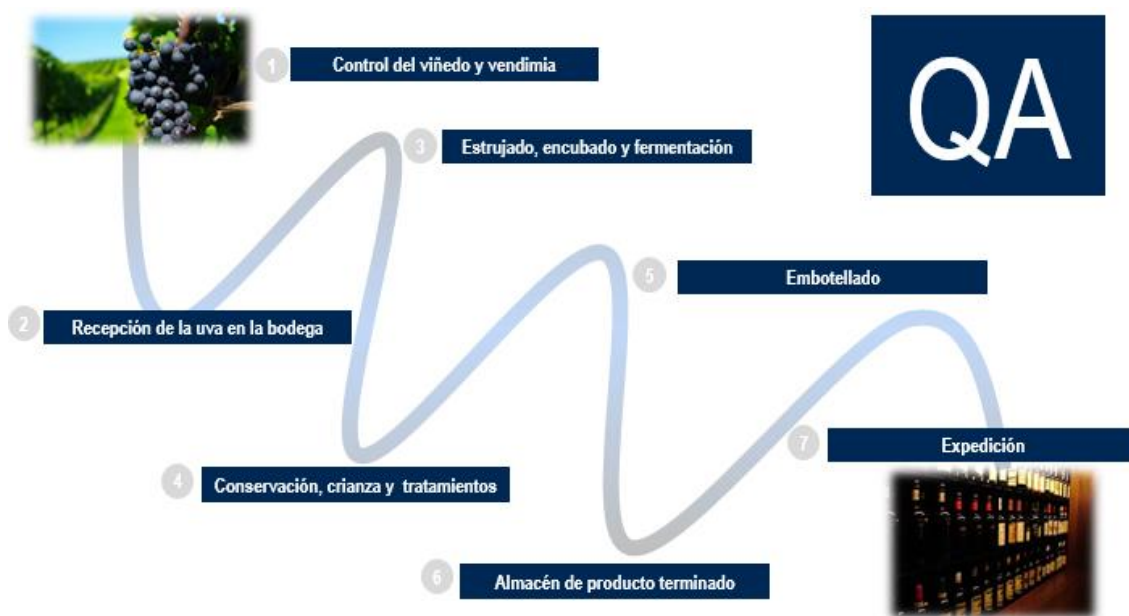


Figura 18 – Proceso completo de la agricultura y producción del vino

6.2.1.2 Objetivos

Si nos ponemos en la piel de un bodeguero, los objetivos que conseguirían que su bodega mejorara serían:

- Eficiencia financiera, que se obtendría a partir de:
 - **Optimización de coste** del proceso
 - **Eficiencia energética** de la bodega
 - **Ahorro en costes** en mantenimiento de maquinaria
 - **Mejor aprovechamiento** de recursos humanos y técnicos de la bodega.
 - **Programación Energética** por adelantado en base a previsiones de consumo.
 - **Optimización** de recursos humanos
- Mejora de calidad, a través de:
 - **Seguimiento ONLINE** del PROCESO COMPLETO Y UNIFICADO.
 - **Fácil seguimiento** de producto y proceso
 - **Rapidez de actuación**
 - **Disminución de daños** por fallos en la elaboración
 - **Optimización y monitorización de KPIs** en tiempo real
 - **Conjunción de KPI's** económicos, calidad y producción
- Gestión de negocio gracias a:
 - **Optimización de tiempo** de proceso de producción
 - **Gestión de información**
 - Sistema de **Alertas** de seguridad
 - Medición de Proceso en términos de **SLA's**
 - **Históricos** accesibles
 - **Análisis forense y peritaje** de problemas e incidencias.
 - **Estadísticas** completas y unificadas
 - **Mejora de producto** aunando conocimiento de Enólogo y Sistema Integral de M2M

6.2.1.3 Propuesta

Para cumplir con los objetivos establecidos en el punto anterior, se diseña una solución integral que contemple los siguientes puntos:

- **Gestión de la información:** Una vez capturados los datos relacionados con los recursos, las instalaciones y la vida útil del producto/proceso, se gestiona la información y se presenta a través de informes de históricos, estadísticas o provisiones (Ver Figura 19).



Figura 19 – Agricultura y producción inteligente del vino. Gestión de la información

- Monitorización y supervisión: Mediante la monitorización y control de los procesos, y el seguimiento de los sensores que influyen en dichos procesos, se obtiene información valiosa para mantener los niveles más altos de calidad (Ver Figura 20).



Figura 20 - Agricultura y producción inteligente del vino. Monitorización y supervisión.

- Alertas: Las alertas permiten avisar al técnico correspondiente, de situaciones de riesgo que aparezcan durante el proceso, detectadas cuando los valores recogidos por los sensores, se exceden de los límites mínimos y máximos establecidos. En la Figura 21 se pueden observar ejemplos de alertas relacionadas con el caudal de la bomba de

entrada, la temperatura y la humedad de la sala de producción y un histórico de alertas por exceso de temperatura en la sala de producción.

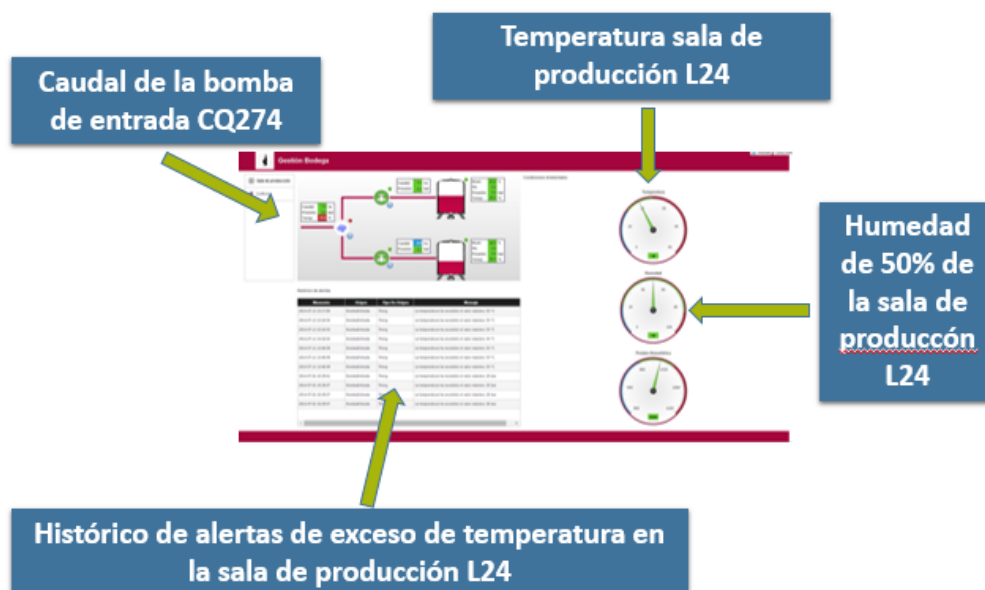


Figura 21 - Agricultura y producción inteligente del vino. Alertas

- Trazabilidad del proceso: Permite seguir el proceso de evolución del producto en cada una de sus etapas lo que permite detectar los puntos críticos del mismo (Ver Figura 22).



Figura 22 - Agricultura y producción inteligente del vino. Trazabilidad del proceso

- Control preventivo: Además de las alertas, es posible realizar un control preventivo para asegurar que todos los elementos y etapas del proceso están dentro de la normalidad, anticipándose a una situación de emergencia (Ver Figura 23).

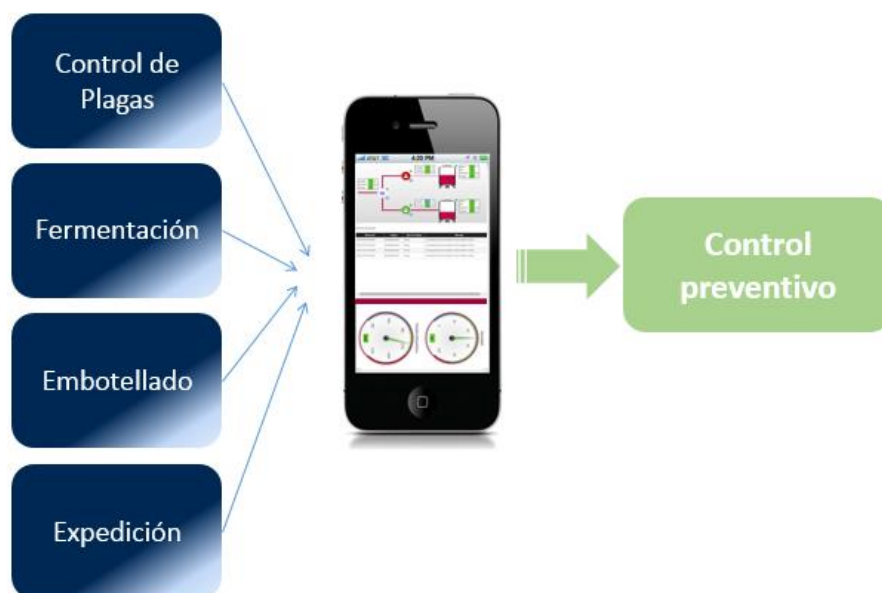


Figura 23 - Agricultura y producción inteligente del vino. Control preventivo

- Mejora continua, a través de la gestión integral del proceso desde el cultivo de la uva: En la Figura 24 se muestra un listado de los parámetros que se desean controlar durante el cultivo de la uva.



Figura 24 - Agricultura y producción inteligente del vino. Agricultura de precisión

En la Figura 25 se muestra un listado de los parámetros que se desean controlar durante la elaboración del vino.



Control de los parámetros que afectan al resultado final:

- Temperatura
- Humedad
- Nivel de oxígeno
- Medición de gases
- Nivel de llenado
- Otra serie de parámetros físico-químicos que puedan resultar de interés



Figura 25 - Agricultura y producción inteligente del vino. Control de parámetros

Y finalizando por la trazabilidad de las botellas de vino y la expedición de las mismas para optimizar el almacenamiento y distribución se ofrece (Ver Figura 26):



Optimización del almacenamiento y distribución

- Conocimiento del estado de una partida en todo momento
- Monitorización del recorrido desde la planta hasta el cliente
- Cálculo de rutas óptimas
- Geoposicionamiento
- Programación de Mantenimientos
- Referencia de pedidos transportados



Figura 26 - Agricultura y producción inteligente del vino. Almacenamiento y distribución

Para conseguir el principal objetivo, obtener una alta calidad del producto a través de un análisis químico de los parámetros recogidos (Ver Figura 27):

Optimización del control de calidad

- Parámetros del estado del producto en las cubas en tiempo real
- Análisis de composición
- Niveles de acidez
- Disolución de iones
- Otra serie de parámetros físico-químicos que puedan resultar de interés



Figura 27 - Agricultura y producción inteligente del vino. Optimización del control de calidad

Para ofrecer esta solución global, y tras la implementación del prototipo, se propondría la arquitectura que aparece en la Figura 28:



Figura 28 - Agricultura y producción inteligente del vino. Arquitectura de la solución.

6.2.1.4 *Simulación de la solución*

Tras explicar el proceso de negocio de la bodega y tener una propuesta concreta para la implantación de la solución vertical, se ha realizado una simulación a partir del prototipo del nodo de sensores que se ha implementado.

Para ello, se han creado los elementos principales que intervienen tanto en las viñas, como en el funcionamiento de una bodega.

El objetivo en este punto no es detallar cómo se ha realizado el desarrollo ni que objetos se han creado como se ha explicado para el prototipo del nodo de sensores en el punto 5.3.1. El foco en este apartado es realizar una simulación de este caso de negocio en concreto. El desarrollo de esta simulación se ha realizado al igual que en el prototipo, creando Things, Things Templates, Thing Shapes, etc. para modelar los objetos reales como puede ser una viña, una bomba, una válvula, un depósito, etc., además de servicios y suscripciones para crear alertas que simulen riesgos en la bodega o las viñas.

Para que el contexto sea lo más real posible, los valores que toman los objetos son datos reales del nodo de sensores. La situación que hay que imaginarse es que en cada uno de esos objetos reales de la bodega y las viñas, existe un nodo de sensores que recoge los parámetros que se quieran medir, dependiendo del objeto que sea.

Por ejemplo, para representar y simular el estado de las viñas, hay que suponer que en cada viña existiría un nodo de sensores como el del prototipo, que recogería los parámetros de Humedad, PH y temperatura, de los cuales, en esta simulación, la humedad y temperatura son datos reales capturados por el nodo físico de sensores.

Para las capas de visualización, se han utilizado las herramientas y widgets que ofrece la plataforma Thingworx, además de incluir imágenes propias para el escenario en concreto.

La capa de visualización creada se divide en dos escenarios. Por un lado se ha querido reflejar la parte exterior de las viñas, en las que se controlen las condiciones ambientales (Temperatura, humedad y presión atmosférica) y el estado de las cepas (humedad, ph y temperatura).

A continuación, en la Figura 29, se muestra dicho escenario:

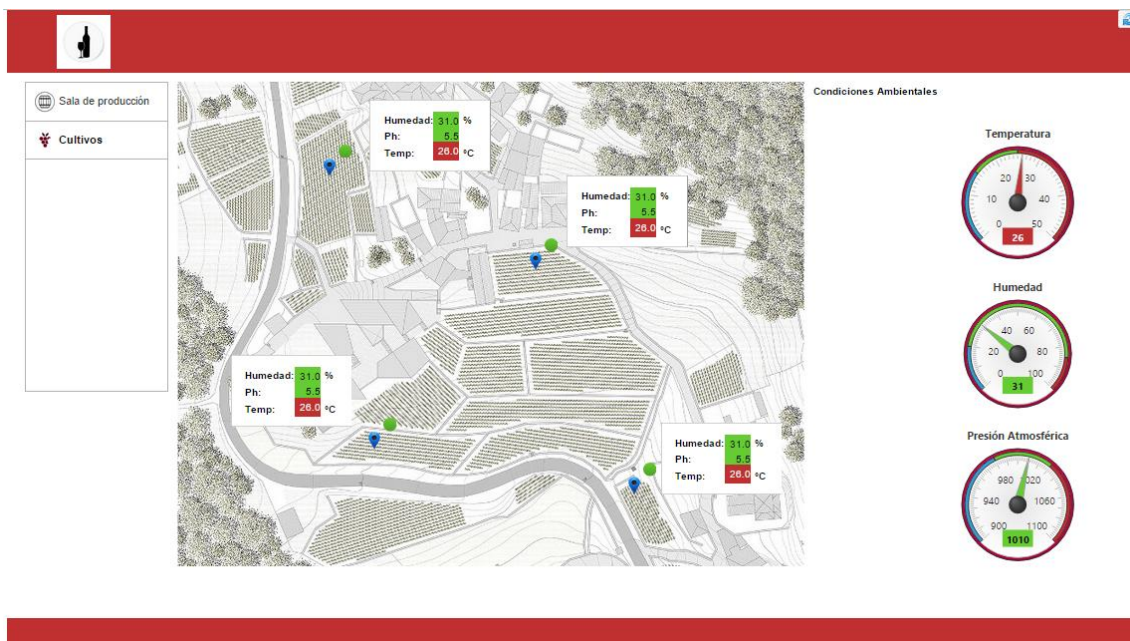


Figura 29 - Agricultura y producción inteligente del vino. Vista cultivos.

Se pueden observar los parámetros que recogerían los nodos de sensores situados en las cepas. Además se han utilizado diferentes colores para que, a simple vista, se pueda ver que hay algún valor fuera del rango establecido y alerte a la persona responsable para que actúe o directamente se active el actuador si existiese.

Por otro lado, se ha desarrollado otro escenario que represente la sala de elaboración del vino de la bodega y que además permita actuar sobre los elementos que intervienen en el proceso: una bomba, dos válvulas y dos depósitos de vino. La Figura 30 muestra una vista de la sala de producción, con los elementos correspondientes.

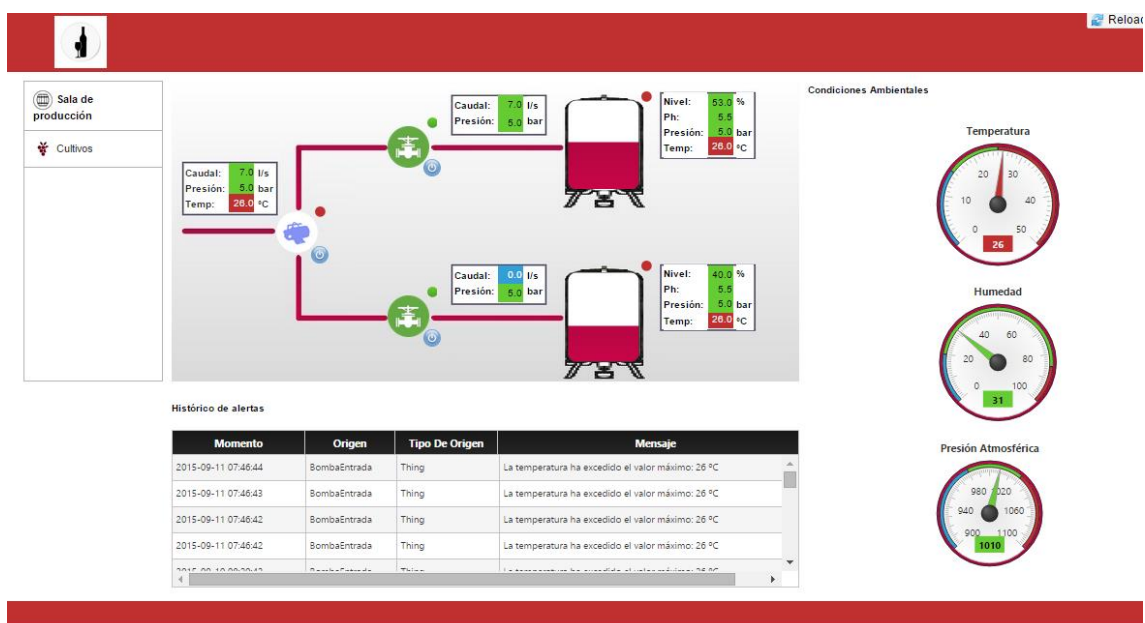


Figura 30 - Agricultura y producción inteligente del vino. Vista sala de producción.

Se ha creado un botón de acción, de modo que desde la aplicación se pueda manejar el proceso y detener o continuar el proceso de elaboración. Si por ejemplo, queremos detener el flujo debido a que la temperatura en los depósitos ha superado la máxima permitida, se pulsa sobre el botón de la bomba y se enviarían los datos correspondientes al actuador situado en la válvula física, para que ésta se cerrara y por tanto se detuviera el proceso.

A diferencia de la imagen anterior, en la Figura 31 se observa cómo gráficamente el flujo ha cambiado de color y ha pasado de tener un relleno de color rojo, a tener un relleno de color blanco que indica que el flujo del vino se ha detenido, además de cambiar el color del icono de la bomba de entrada.

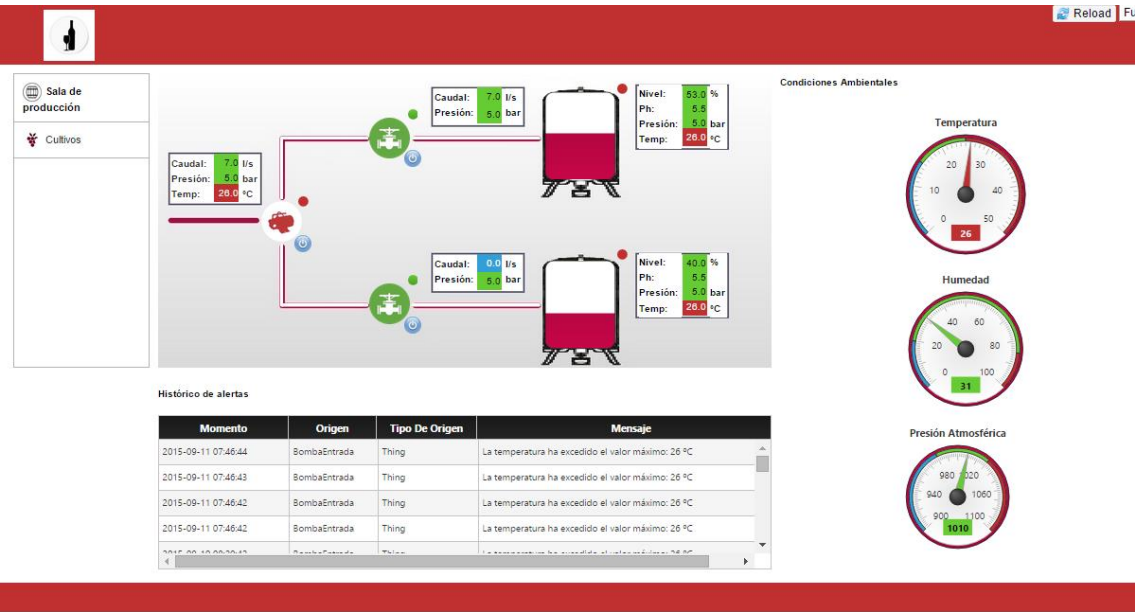


Figura 31 - Agricultura y producción inteligente del vino. Vista sala de producción con proceso detenido

También se ha creado otro elemento de visualización, como el de la Figura 32, para crear un cuadro de mando desde el que pueda observar en detalle, el estado de cada uno de los elementos y permita modificar los rangos máximos y mínimos establecidos para cada propiedad. También es posible configurar el dispositivo para cambiar su estado (en funcionamiento o en mantenimiento), así como activar o desactivar la opción de alarma en caso de incidencia:

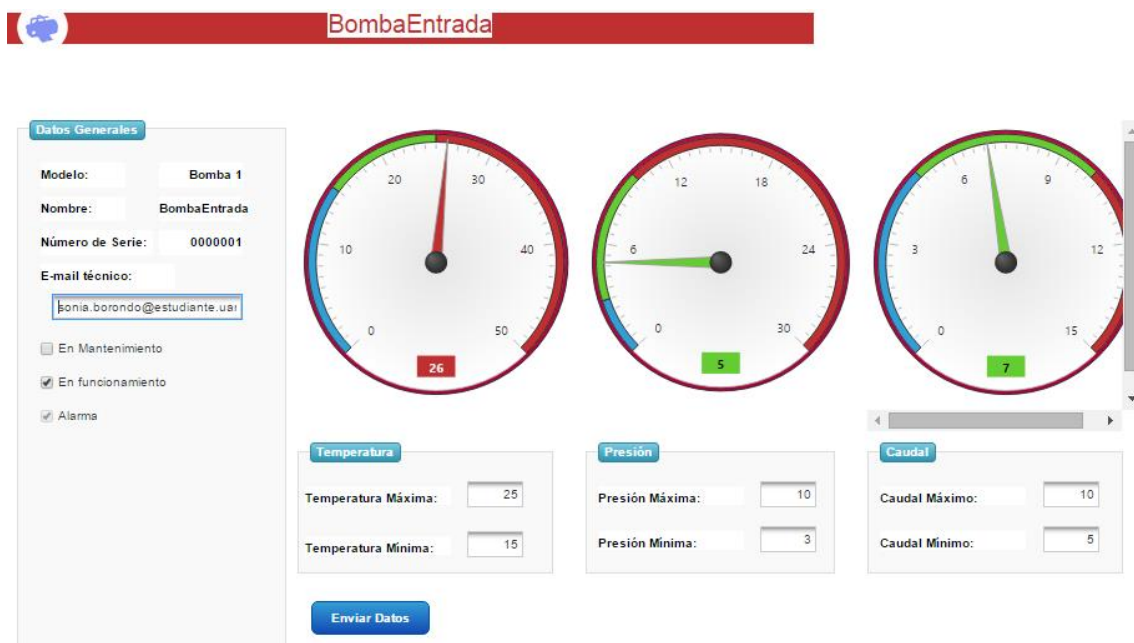


Figura 32 - Agricultura y producción inteligente del vino. Cuadro de mando de la bomba de la sala de producción.

6.2.2 Sistema de telemetría de boyas y estaciones meteorológicas

6.2.2.1 Escenario actual

A diferencia del caso de negocio anterior, cuyo objetivo era fundamentalmente mejorar la calidad de un producto (en ese caso, el vino), en esta simulación los objetivos se enfocan más a evitar posibles catástrofes, proporcionar un punto de información a navegantes, realizar estudios meteorológicos, etc.

Tras realizar un estudio de negocio para aprender cómo funciona una estación meteorológica o una boya marítima, el proceso de telemetría sería el que se muestra en la Figura 33.



Figura 33 - Proceso del sistema de telemetría de boyas y estaciones meteorológicas

6.2.2.2 Objetivos

Los objetivos que se persiguen implantando esta solución vertical, serían los mismos que en el caso de la agricultura y producción inteligente del vino, pero aplicados en este contexto:

- Eficiencia financiera, que se obtendría a partir de:
 - **Optimización de coste** del sistema
 - **Eficiencia energética** del sistema
 - **Ahorro en costes** en mantenimiento de maquinaria
 - **Mejor aprovechamiento** de recursos humanos y técnicos.
 - **Programación Energética** por adelantado en base a previsiones de consumo.
 - **Optimización** de recursos humanos
- Mejora de calidad, a través de:
 - **Seguimiento ONLINE de la situación meteorológica y oceanográfica.**
 - **Fácil seguimiento y monitoreo** de las boyas y las estaciones.
 - **Rapidez de actuación.**
 - **Aumento de seguridad** por fallos en los sistemas.
 - **Optimización y monitorización de KPIs** en tiempo real.
 - **Conjunción de KPI's** económicos, calidad y gestión.
- Gestión de negocio gracias a:
 - **Optimización de tiempo** de proceso de actuación
 - **Gestión de información**
 - Sistema de **Alertas** de seguridad
 - Medición de Proceso en términos de **SLA's**
 - **Históricos** accesibles

- **Análisis forense y peritaje** de problemas e incidencias.
- **Estadísticas** completas y unificadas
- **Mejora de procedimiento de actuación** aunando conocimiento de expertos y Sistema Integral de M2M.

6.2.2.3 Propuesta

Para cumplir con los objetivos establecidos en el punto anterior, se establece una solución integral que contemple los siguientes puntos:

- **Gestión de la información:** La información proveniente de los recursos, las instalaciones y la vida útil del producto/proceso, se gestiona a través de informes de históricos, estadísticas o provisiones.
- **Monitorización y supervisión:** Mediante la monitorización y supervisión de las boyas y estaciones meteorológicas, se obtiene información valiosa para realizar estudios meteorológicos o por ejemplo, para mejorar el aprovechamiento energético a través del oleaje. (Ver Figura 34).



Figura 34 - Telemetría de boyas y estaciones meteorológicas. Monitorización y supervisión.

- **Alertas:** Las alertas permiten avisar al usuario o técnico correspondiente de posibles situaciones de riesgo dadas, que se detectan cuando los valores recogidos por los sensores se exceden de los límites mínimos y máximos establecidos. En la Figura 35 se muestra alertas sobre el estado de la humedad, temperatura y presión, cuyos valores han excedido los límites establecidos.

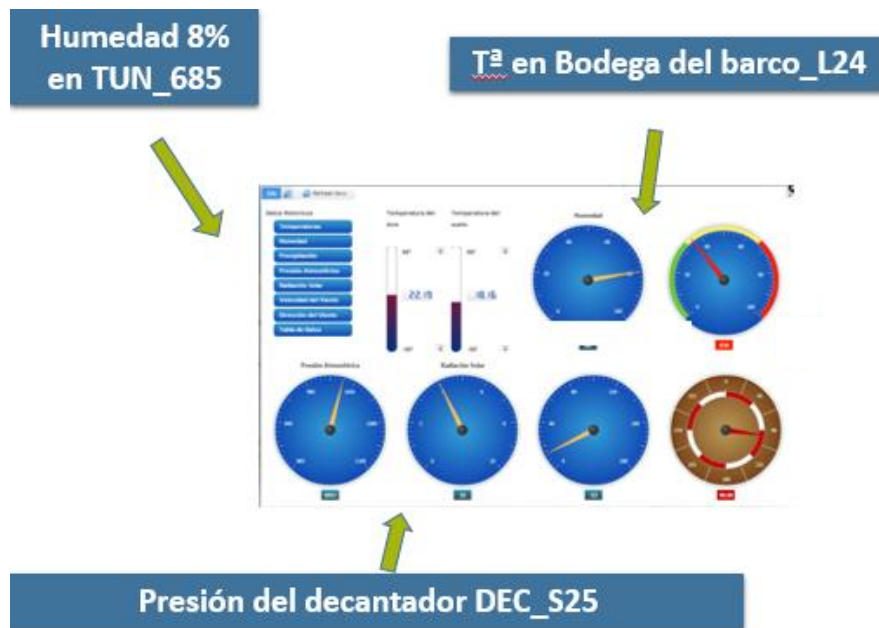


Figura 35 - Telemetría de boyas y estaciones meteorológicas. Alertas.

- Trazabilidad del proceso: Permite disponer en tiempo real de los datos meteorológicos, que son analizados y visualizados para aportar información útil. (Ver Figura 36).



Figura 36 - Telemetría de boyas y estaciones meteorológicas. Trazabilidad del proceso

- Control preventivo: Además de las alertas, es posible realizar un control preventivo para asegurar que todos los elementos y etapas del proceso están dentro de la normalidad, anticipándose a una situación de emergencia o catástrofe (Ver Figura 37).

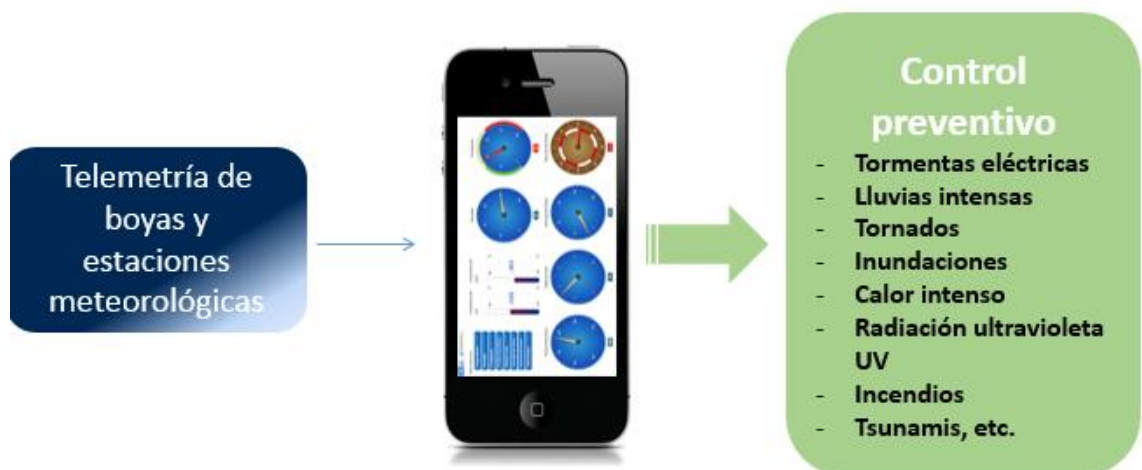


Figura 37 - Telemetría de boyas y estaciones meteorológicas. Control preventivo.

Para ofrecer esta solución global, y tras la implementación del prototipo, se propondría la siguiente arquitectura:



Figura 38 - Telemetría de boyas y estaciones meteorológicas. Arquitectura de la solución.

6.2.2.4 Simulación de la solución

Como para el caso de los viñedos y la bodega, se han creado los elementos principales que intervienen tanto en una estación meteorológica, como en un sistema de boyas.

El desarrollo de esta simulación se ha realizado al igual que en el prototipo, creando Things, Things Templates, Thing Shapes, etc. para modelar los objetos reales como puede ser una boya o una estación meteorológica, además de servicios y suscripciones para crear alertas que simulen riesgos meteorológicos.

Al igual que en el caso anterior, para que el contexto sea lo más real posible, los valores que toman los objetos son datos reales del nodo de sensores. La situación que hay que imaginarse es que en cada boya o estación meteorológica existen nodos de sensores que recogen los parámetros que se quieran medir, dependiendo del objeto que sea.

Por ejemplo, para representar y simular las condiciones meteorológicas de una estación, hay que suponer que en ella habría diferentes nodos con diferentes sensores como el del prototipo, que recogería los parámetros de precipitación, humedad, temperatura, presión atmosférica, radiación solar, etc. de los cuales, en esta simulación, la humedad y temperatura son datos reales capturados por el nodo físico de sensores.

Para las capas de visualización, se han utilizado las herramientas y widgets que ofrece la plataforma Thingworx, además de incluir imágenes propias para el escenario en concreto.

La capa de visualización creada se divide en dos escenarios. Por un lado se ha querido reflejar las estaciones meteorológicas, en las que se miden las condiciones ambientales y se muestra gráficamente la situación de dichas boyas.

A continuación se muestra dicho escenario:

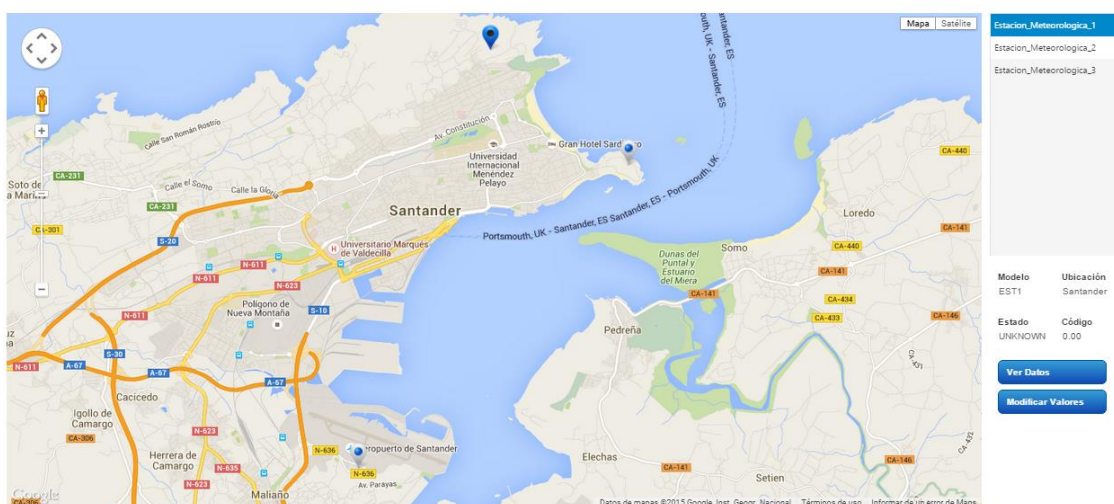


Figura 39 – Telemetría de boyas y estaciones meteorológicas. Vista estaciones.

Se pueden observar los detalles de los parámetros que recogerían los sensores situados en las estaciones a través de cuadro de mando creado que aparece si se pincha sobre una estación en concreto y se pulsa el botón “Ver Datos”.

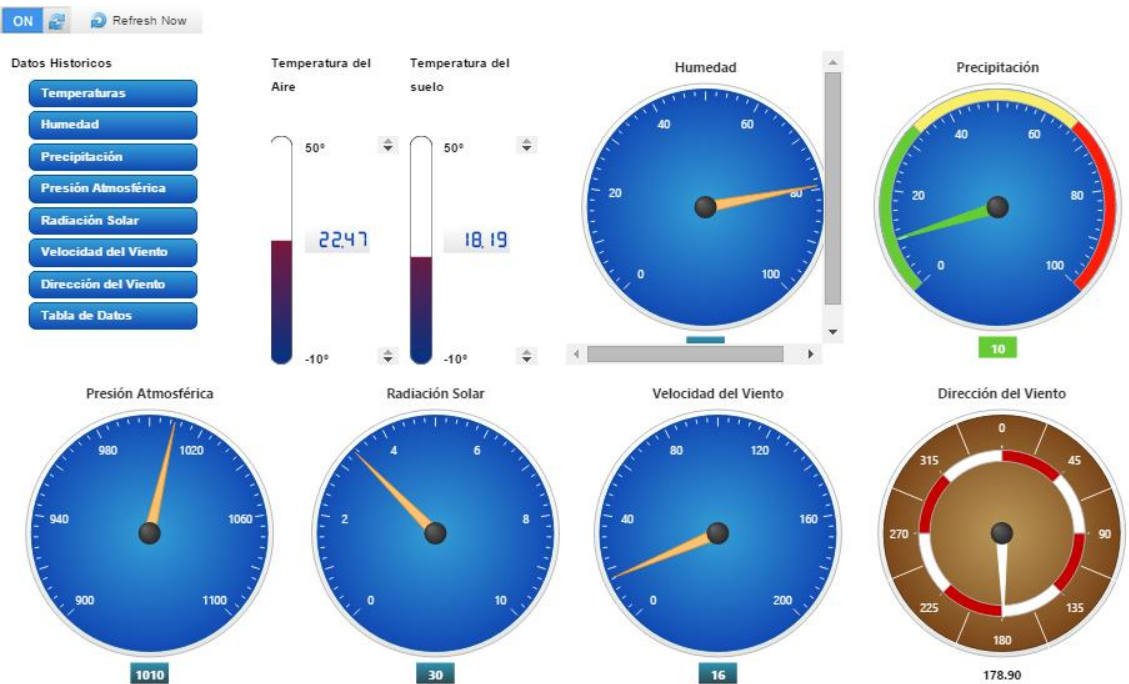


Figura 40 - Telemetría de boyas y estaciones meteorológicas. Vista cuadro de mando estación.

Además se han utilizado diferentes colores para que, a simple vista, se pueda ver que hay algún valor fuera del rango establecido y alerte sobre alguna posible catástrofe a la persona responsable para que actúe o directamente se active el actuador si existiese.

Por otro lado, se ha desarrollado otro escenario que represente la ubicación de las boyas y ver en detalle el estado de cada una de ellas:

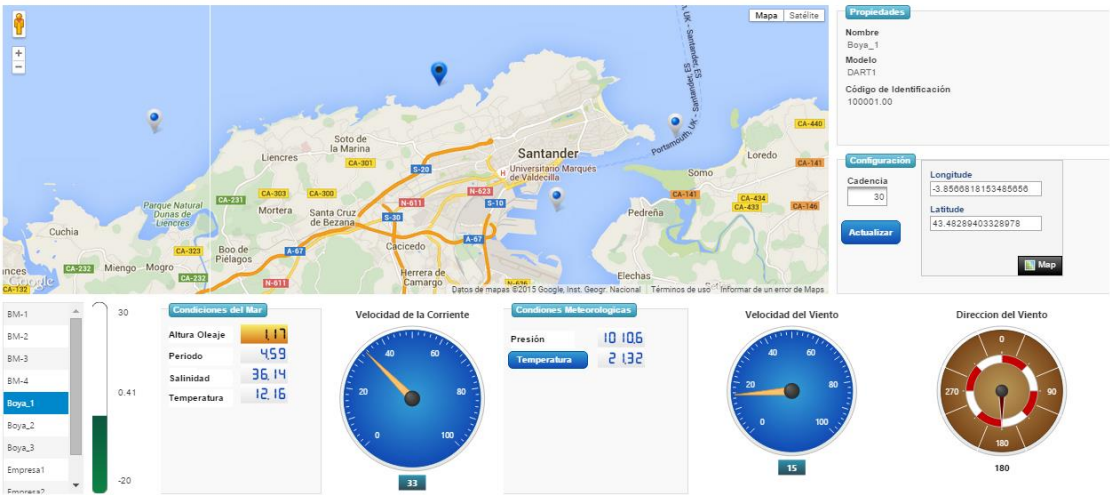


Figura 41 - Telemetría de boyas y estaciones meteorológicas. Vista boyas.

7 Conclusiones y trabajo futuro

El objetivo fundamental de este trabajo era el desarrollo de una nueva solución integral para la Internet of Things que integrara los parámetros recogidos por dispositivos físicos a través de sensores en una aplicación de usuario en la que éste pueda gestionar y monitorizar dichos dispositivos, y que sirviera de base para futuras implementaciones de soluciones verticales en este contexto.

Este objetivo ha sido satisfecho. En una primera fase del trabajo se llevó a cabo una importante recopilación bibliográfica y revisión teórica sobre los aspectos básicos relacionados con el tema de la Internet of Things. En esta fase se descubrió, entre otras cosas, la existencia de un novedoso protocolo de red de comunicaciones, Sigfox, que permite conectar los objetos a la nube. Este protocolo ha sido aplicado exitosamente a la solución propuesta.

En esta memoria se ha descrito detalladamente el contexto de este trabajo y las decisiones tomadas para resolver el reto propuesto, y se ha presentado el diseño e implementación de la solución integral desarrollada, que abarca desde la recepción de las señales que transmite un dispositivo físico, pasando por el desarrollo del modelo o ecosistema, hasta la capa de visualización y monitorización de la información por parte del usuario.

El trabajo desarrollado constituye una base para que, en cualquier caso de negocio, se pueda implementar una solución IoT vertical enfocada a hacer eficientes, optimizar y gestionar estos ámbitos o negocios concretos. Esto ha quedado ilustrado mediante dos ejemplos concretos pertenecientes a dos contextos muy diferentes, que han sido descritos detalladamente en esta memoria: el de la agricultura y producción inteligente del vino y el de un sistema de telemetría de boyas y estaciones meteorológicas.

Cabe destacar por un lado el ahorro de costes energéticos y económicos que supone aplicar soluciones de negocio basadas en IoT. En particular, la solución propuesta permite, por una parte, incrementar este ahorro al ajustar los requisitos de transmisión de mensajes a un tamaño limitado y estrictamente necesario, para poder llevar a cabo la transmisión a través de Sigfox. Por otra parte, cabe matizar el ahorro de tiempo que supone la implementación de la aplicación IoT mediante el desarrollo basado en modelos que permite ThingWorx.

Como trabajo futuro, se plantea, por un lado la posibilidad de la implantación real de la solución de la agricultura y producción del vino inteligente, tras contactar con varios bodegueros interesados en recibir una demostración del ejemplo de esta solución. Por otro lado, también se quieren ampliar el número de ejemplos, para abarcar otros sectores o entornos concretos. Por ejemplo, realizar otro ejemplo para un parking inteligente y otro para la gestión y monitorización del suministro de cerveza.

8 Bibliografía

- [1] Peter Waher (2015) Learning internet of things: explore and learn about internet of things with the help of engaging and enlightening tutorials designed for raspberry Pi. Packt Publishing
- [2] Kevin Ashton. That 'Internet of Things' Thing - RFID Journal. [Online. Ultimo acceso: 11/09/2015]. <http://www.rfidjournal.com/article/print/4986>.
- [3] Michael E. Porter and James E. Heppelmann (2014). How Smart, Connected Products Are Transforming Competition. Harvard Business Review. Harvard Business School Publishing, November 2014.
- [4] Hersent, Olivier; Boswarthick, David; Elloumi, Omar (2012) The internet of things: key applications and protocols. Wiley; 2 edition
- [5] Tecnología Sigfox [Online]. <http://www.sigfox.com/es/#!/technology>
- [6] Módulo de conectividad con red Sigfox. [Online Ultimo acceso: 11/09/2015]]. <http://www.tst-sistemas.es/Docs/sigfox.pdf>
- [7] Herramienta de diseño para aplicaciones M2M, control remoto y monitorización inalámbrica. [Online]. <http://www.tst-sistemas.es/Docs/TSmoTe.pdf>
- [8] API & Callbacks Backend Sigfox. [Online]. <http://makers.sigfox.com/#api>
- [9] ThingWorx 4.2 Fundamentals Workbook.
- [10] ThingWorx Mashup Builder Quickstart Guide.
- [11] Jeff Desjardins. The present and the future of Internet of Things [Online Ultimo acceso: 11/09/2015]. <http://www.visualcapitalist.com/present-future-internet-things/>
- [12] Y Zhang, R Yu, S Xie, W Yao, Y Xia (2011) Home M2M networks: architectures, standards, and QoS improvement. Communications Magazine, IEEE (Volume:49 , Issue: 4). Pags. 44-52.
- [13] Michael E. Porter (2015). Internet of Things to Change Competitive Landscape Says Strategy [Online Último acceso: 11/09/2015] <http://www.slideshare.net/KnudLasseLueth/global-io-t-day-2015-vslideshare-46925168>
- [14] Castellani, A.P. Bui, N. ; Casari, P. ; Rossi, M.(2010) Architecture and protocols for the Internet of Things: A case study [Online Último acceso: 11/09/2015]. <http://webofthings.org/wot/2010/pdfs/144.pdf>
- [15] J Gubbi, R Buyya, S Marusic, M Palaniswami. (2013) Internet of Things (IoT): A vision, architectural elements, and future directions. [Online último acceso: 11/09/2015] <http://www.sciencedirect.com/science/article/pii/S0167739X13000241>
- [16] S Tozlu, M Senel, W Mao (2012) Wi-Fi enabled sensors for internet of things: A practical approach. IEEE Communications Magazine, Volume 50, Pags. 134-143.
- [17] N Mitton, D Simplot-Ryl (2011) From the Internet of things to the Internet of the physical world. Comptes Rendus Physique, vol. 7.
- [18] C Floerkemeier (2008). The Internet of Things: First International Conference, IOT 2008, Zurich, Switzerland.